

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-57521

(P2001-57521A)

(43)公開日 平成13年2月27日(2001.2.27)

(51)Int.Cl.<sup>7</sup>

識別記号

F I

テーマコード\*(参考)

H 0 3 M 13/27

H 0 3 M 13/27

13/15

13/15

13/23

13/23

13/29

13/29

H 0 4 J 13/00

H 0 4 J 13/00

A

審査請求 未請求 請求項の数16 OL 外国語出願 (全 34 頁)

(21)出願番号 特願2000-109059(P2000-109059)

(71)出願人 390023157

(22)出願日 平成12年4月11日(2000.4.11)

ノーテル・ネットワークス・リミテッド  
NORTEL NETWORKS LIM  
ITED

(31)優先権主張番号 2 2 6 8 8 5 3

カナダ国 エッチ・2・ワイ 3・ワイ・

(32)優先日 平成11年4月13日(1999.4.13)

4 ケベック州 モントリオール セイン

(33)優先権主張国 カナダ (CA)

ト アントワヌ ストリート ウェスト

380 ワールドトレード センター

オブ モントリオール エイスフロアー

(74)代理人 100081721

弁理士 岡田 次生 (外4名)

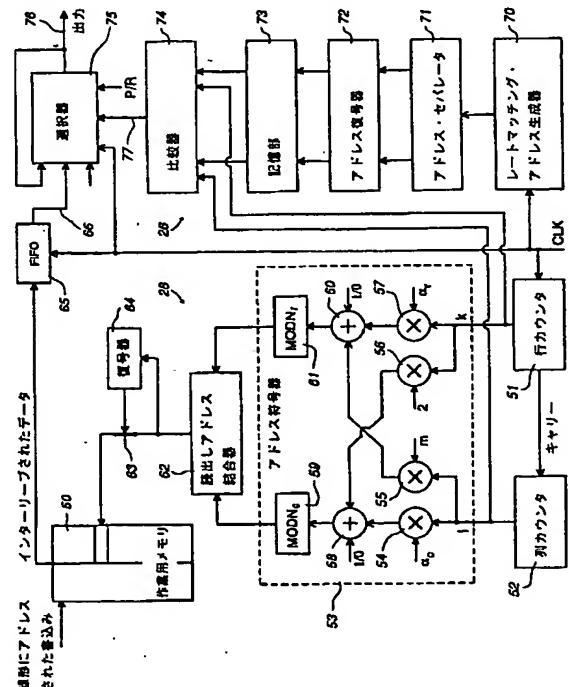
最終頁に続く

(54)【発明の名称】 通信システムのためのレートマッチングおよびチャネル・インターリーピング

(57)【要約】 (修正有)

【課題】 通信システムのレートマッチング方法において、インターリーブされたデータビットのマトリクスにおけるデータビット・レートを、簡単な方法で所望のレートに整合させる。

【解決手段】 所望のデータレートを提供するために削除または反復すべきビットのパターンを、インターリーブされていないデータビットのマトリクスにおいて求めるステップと、インターリーピング処理とは反対の方法で該パターンのそれぞれのビットのアドレスを復号し、インターリーブされたデータビットのマトリクスにおけるビットのそれぞれのアドレスを生成するステップと、該それぞれのアドレスに従ってインターリーブされたデータビットのそれぞれのビットを削除または反復するステップとを含む。



## 【特許請求の範囲】

【請求項 1】 予め決められたインターリーブ処理によってインターリーブされたデータビットのマトリクスにおけるデータビットのレートを、該マトリクスから導き出された冗長なデータビットの削除またはデータビットの反復により、所望のデータビットのレートに整合させる方法であって、

前記データビットのインターリーブされていないマトリクスにおいて、前記所望のデータレートを提供するために削除または反復されるべきビットのパターンを求めるステップと、

インターリーブ処理とは反対の方法で、前記パターンのそれぞれのビットのアドレスを復号し、インターリーブされたデータビットのマトリクスのビットのそれぞれのアドレスを生成するステップと、

前記それぞれのアドレスに従って、前記インターリーブされたデータビットのそれぞれのビットを削除または反復するステップと、

を含むデータビット・レート在所望のデータビット・レ

行並べ替え  $I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$

列並べ替え  $I_c(l) = [\alpha_c k + f_r(k)] \bmod N_c$

ここで  $I_r(k)$  は行インデックス  $k$  のデータビットを表し、 $k$  は  $1 \sim N_r$  の整数であり、 $\alpha_r$  は整数であり、 $f_c(l)$  は列インデックス  $l$  のゼロでない関数であり、 $l$  は  $1 \sim N_c$  の整数であり、 $I_c(l)$  は列インデックス  $l$  のデータビットを表し、 $\alpha_c$  は整数であり、 $f_r(k)$  はゼロまたは行インデックス  $k$  の関数であり、 $\bmod N_r$  および  $\bmod N_c$  はモジュロ  $N_r$  およびモジュロ  $N_c$  算術をそれぞれ表しており、

インターリーブされたデータビットが列ごとにマトリクスから導き出される請求項 1 から請求項 3 のいずれかに記載のデータビット・レートを所望のデータビット・レートにマッチングさせる方法。

【請求項 5】 前記  $f_c(l)$  が、 $f_c(l) = ml + [N_r + 1] \bmod 2$  であり、 $m$  は整数である請求項 4 に記載のデータビット・レートを所望のデータビット・レートにマッチングさせる方法。

【請求項 6】 前記  $m$  が、およそ  $N_r/N_c$  に等しい請求項 5 に記載のデータビット・レートを所望のデータビツ※40

行並べ替え  $I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$

列並べ替え  $I_c(l) = [\alpha_c k + f_r(k)] \bmod N_c$

ここで、 $I_r(k)$  は行インデックス  $k$  のデータビットを表し、 $k$  は  $1 \sim N_r$  の整数であり、 $\alpha_r$  は整数であり、 $f_c(l)$  は列インデックス  $l$  のゼロでない関数であり、 $l$  は  $1 \sim N_c$  の整数であり、 $I_c(l)$  は列インデックス  $l$  のデータビットを表し、 $\alpha_c$  は整数であり、 $f_r(k)$  はゼロまたは行インデックス  $k$  の関数であり、 $\bmod N_r$  および  $\bmod N_c$  はモジュロ  $N_r$  およびモジュロ  $N_c$  算術

＊ートにマッチングさせる方法。

【請求項 2】 前記アドレスの復号化が、前記データビットのインターリーブされていないマトリクスからインターリーブされたデータビットを生成するアドレスの符号化と同じ方法で実行される請求項 1 に記載のデータビット・レートを所望のデータビット・レートにマッチングさせる方法。

【請求項 3】 前記削除または反復されるべきビットのパターンが、削除または反復されるべきビット数およびマトリクスの列インデックスに依存する請求項 1 または請求項 2 に記載のデータビット・レートを所望のデータビット・レートにマッチングさせる方法。

【請求項 4】 前記インターリーブ処理が、 $N_r$  個の行および  $N_c$  個の列のマトリクスの行および列の並べ替えを含み、該並べ替えにおいて、インターリーブされるべきデータビットが、以下の式 (1) に従って行ごとに表され、

【数 1】

式 (1)

※ト・レートにマッチングさせる方法。

【請求項 7】 前記  $f_r(k)$  が、 $f_r(k) = 2k + [N_c + 1] \bmod 2$  である請求項 4 から請求項 6 に記載のデータビット・レートを所望のデータビット・レートにマッチングさせる方法。

【請求項 8】 前記  $\alpha_r$  が、 $N_r / \log_2(\log_2(N_r))$  より小さい最大素数である請求項 4 から請求項 7 に記載のデータビット・レートを所望のデータビット・レートにマッチングさせる方法。

【請求項 9】 請求項 1 から請求項 8 のいずれかに記載の方法を実行するよう構成されたレートマッチング装置。

【請求項 10】  $N_r$  個の行および  $N_c$  個の列のマトリクスの行および列を並べ替えることを含むデータビットをインターリーブする方法であって、インターリーブされるべきデータビットは、以下の式 (2) に従って行ごとに表され、

【数 2】

式 (2)

をそれぞれ表しており、インターリーブされたデータビットが列ごとにマトリクスから導き出されるデータビットをインターリーブする方法。

【請求項 11】 前記  $m$  が、およそ  $N_r/N_c$  に等しい請求項 10 に記載のデータビットをインターリーブする方法。

【請求項 12】前記  $\alpha_r$  が、 $N_r / \log_2 (\log_2 (N_r))$  より小さい最大素数である請求項 10 または請求項 11 に記載のデータビットをインターリーブする方法。

【請求項 13】請求項 10、11 または 12 に記載の方法を実行するよう構成されたデータ・インターリーバ。

【請求項 14】符号化されたデータビットを削除することにより、並列接続畳み込み符号化されたデータをインターリーブしてレートマッチングする方法であって、該符号化されたデータは、組織ビットおよびパリティビットを有しており、

前記パリティビットとは別個に組織ビットをインターリーブするステップと、

前記インターリーブされたパリティビットからパリティビットを削除してレートマッチングを提供するステップと、

を含む並列接続畳み込み符号化されたデータをインターリーブしてレートマッチングする方法。

【請求項 15】符号化されたデータビットを削除することにより、並列接続畳み込み符号化されたデータをインターリーブしてレートマッチングする方法であって、該符号化されたデータは、組織ビットおよびパリティビットを有しており、

前記組織ビットとは別個にパリティビットをインターリーブするステップと、前記インターリーブされた組織ビットの組織ビットの任意の反復より大きい反復ファクターで、前記インターリーブされたパリティビットのパリティビットを反復し、レートマッチングを提供するステップと、

を含む並列接続畳み込み符号化されたデータをインターリーブしてレートマッチングする方法。

【請求項 16】請求項 14 または 15 に記載された方法を実行するよう構成された符号化インターリービングおよびレートマッチング装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、通信システムのためのレート・マッチングおよびチャネル・インターリーピングに関する。

【0002】

【従来の技術】通信システムにおいて、前方誤り訂正 (FEC) を使用してデータのインターリーピング (interleaving) を実行することにより、デインターリーピング (deinterleaving) の際に、エラーを分散させてそれらの訂正を容易にすることは周知である。典型的には、そのようなインターリーピングは、データブロックをインターリーブするブロック・インターリーバを使用する。いわゆるターボ符号化 (並列接続畳み込み符号化) は、インターリーピングの前後で入力データからそれぞれのパリティビットを生成する 2 つの畳み込み符号器への入力間に置かれたインターリーバを使用する。特

に無線通信システムにおいて、ターボ符号化の使用が注目されてきており、よってインターリーバの形態も注目されてきている。

【0003】また、いわゆる第 3 世代の CDMA (符号分割多重アクセス) 無線通信システムでは、典型的には 10 ms の無線フレーム持続時間に対応するブロックにおいてデータをインターリーブする (すなわち順序を並べ替える) よう動作するチャネルすなわちフレーム間インターリーバを必要とする無線通信システムが開発されてきている。そのようなシステムでは、チャネル・インターリーバは、レートマッチング機能の前または後ろのいずれかに設けられる。ここで、レートマッチング機能は、様々なデータレートが無線フレームレートに整合 (マッチング) するよう、および典型的にはデータシンボル (このケースではデータビット) のパンクチャリング (puncturing: 切り捨て) または反復 (repetition) を実行するよう動作する。

【0004】

【発明が解決しようとする課題】実現するのが容易であり、かつフレームサイズ、フレーム数およびパンクチャリング率のような変数から比較的独立した方法で、デインターリーブされたフレームにおいて、パンクチャリングまたは反復されたビット間の距離ができるだけ大きく、パンクチャリングまたは反復されたビットができるだけ均一に分散されるのが望ましい。

【0005】1999 年 3 月 19 日に Wen Tong 他により出願された、「データ・インターリーバおよびデータをインターリーブする方法 (Data Interleaver And Method Of Interleaving Data)」と称するカナダの特許出願

(ファイル番号: 10378R0) は、上記に参照したチャネル・インターリーピングを提供するのに有利に使用することができるデータをインターリーブする方法およびデータ・インターリーバについて記述している。この発明は、そのようなチャネル・インターリーピング後のデータについて特定の有利さで使用するのことができるやり方のレートマッチングに関するが、これはまた、他の形式のインターリーブされたデータに対しても適用することができる。また、この発明は、そのようなチャネル・インターリーピングに改良を加え、さらにそのようなチャネル・インターリーピングのアプリケーションに改良を加えるものである。

【0006】

【課題を解決するための手段】この発明の 1 つの側面によると、この発明は、予め決められたインターリーピング処理によってインターリーブされたデータビットのマトリクスにおけるデータビットのレートを、該マトリクスから導き出された冗長なデータビットの削除またはデータビットの反復によって、所望のレートに整合させる方法を提供する。この方法は、該所望のデータレートを提供するために削除または反復されるべきビットのパタ

5

ーンを、前記データビットのインターリーブされていないマトリクスにおいて求めるステップと、前記インターリーブ処理とは反対の方法で前記パターンそれぞれのビットのアドレスを復号し、前記インターリーブされたデータビットのマトリクスにおけるビットのそれぞれのアドレスを生成するステップと、前記それぞれのアドレスに従って、前記インターリーブされたデータビットのそれぞれのビットを削除または反復するステップとを含む。

【0007】アドレスの復号化を、前記データビットのインターリーブされていないマトリクスからインターリーブされたデータビットを生成するアドレスの符号化と同じ方法で実行するのが特に有利であり、かつ実際上必要となりうる。これは、 $N_r$  行および  $N_c$  列のマトリクスの行および列の順番を並べ替えることを含むインターリーブ処理により、この発明の方法の好ましい実施形態で容易化される。ここで、インターリーブされるべきデータビットは、以下の式に従って行ごとに表される。

【0008】

【数3】

行並べ替え  $I_r(k) = [\alpha_r k + f_c(1)] \bmod N_r$

列並べ替え  $I_c(l) = [\alpha_c k + f_r(k)] \bmod N_c$

【0009】ここで、 $I_r(k)$  は、行インデックス  $k$  のデータビットを表し、 $k$  は  $1 \sim N_r$  の整数である。 $\alpha_r$  は整数であり、 $f_c(1)$  は、列インデックス  $1$  のゼロでない関数であり、 $l$  は  $1 \sim N_c$  の整数である。 $I_c(l)$  は、列インデックス  $l$  のデータビットを表す。 $\alpha_c$  は整数であり、 $f_r(k)$  は、ゼロまたは行インデックス  $k$  の関数である。 $\bmod N_r$  および  $\bmod N_c$  は、それぞれモジュロ  $N_r$  およびモジュロ  $N_c$  算術を表す。こうして、インターリーブされたデータビットは、列ごとにマトリクスから導き出される。

【0010】 $f_c(1)$  および  $f_r(k)$  を、 $f_c(1) = m + [N_r + 1] \bmod 2$  (ここで、 $m$  は整数であり、 $N_r / N_c$  におよそ等しい)、 $f_r(k) = 2k + [N_c + 1] \bmod 2$  のように選択し、また  $\alpha_r$  を、 $N_r / \log_2(\log_2(N_r))$  より小さい最大素数として選択するのが最適であると考えられる。また、この発明は、上記引用した方法を実行するよう構成されたレートマッチング装置を提供する。

【0011】この発明の他の側面は、 $N_r$  行および  $N_c$  列のマトリクスの行および列を並べ替えることを含むデータビットをインターリーブする方法を提供する。ここで、インターリーブされるべきデータビットは以下の式に従って行ごとに表される。

【0012】

【数4】

行並べ替え  $I_r(k) = [\alpha_r k + f_c(1)] \bmod N_r$

列並べ替え  $I_c(l) = [\alpha_c k + f_r(k)] \bmod N_c$

6

【0013】ここで、 $I_r(k)$  は行インデックス  $k$  のデータビットを表し、 $k$  は  $1 \sim N_r$  までの整数である。 $\alpha_r$  は整数であり、 $f_c(1) = m + [N_r + 1] \bmod 2$  は列インデックス  $1$  のデータビットを表す。 $\alpha_c$  は整数であり、 $f_r(k) = 2k + [N_c + 1] \bmod 2$  である。 $\bmod 2$ 、 $\bmod N_r$  および  $\bmod N_c$  は、それぞれモジュロ  $2$ 、モジュロ  $N_r$  およびモジュロ  $N_c$  算術を表す。こうして、インターリーブされたデータビットは、列ごとにマトリクスから導き出される。

【0014】また、この発明は、この方法を実行するよう構成されたデータ・インターリーブを提供する。

【0015】この発明の他の側面は、符号化されたデータビットを削除することにより、並列接続畳み込み符号化されたデータをインターリーブしてレートマッチングする方法を提供する。符号化されたビットは、組織ビットおよびパリティビットを有する。

【0016】この方法は、パリティビットとは別個に組織ビットをインターリーブするステップと、該インターリーブされたパリティビットからパリティビットを削除してレートマッチングを提供するステップとを含む。

【0017】この発明のさらなる側面は、組織ビットおよびパリティビットを有する符号化されたデータビットの反復により、並列接続畳み込み符号化データをインターリーブしてレートマッチングする方法を提供する。この方法は、パリティビットとは別個に組織ビットをインターリーブするステップと、インターリーブされたパリティビットを、インターリーブされた組織ビットの任意の組織ビット反復より大きい反復ファクターで反復し、レートマッチングを提供するステップと、を含む。

【0018】さらにこの発明は、これらの方法を実行するよう構成された符号化、インターリーブおよびレートマッチングの装置を提供する。

【0019】さらにこの発明の他の側面は、図4を参照して以下に説明する方法で、インターリーブされレートマッチングされたデータストリームをシャッフリング(shuffle)する方法、および2より多いそのようなデータストリームに対するこの方法の帰納的な適用に関する。

【0020】この発明のさらなる側面は、一般的、実質的または本質的に、図3から図5を参照して以下に説明する方法または装置に関する。

【0021】

【発明の実施の形態】図1を参照すると、第3世代CDMA無線通信システムにおけるサービス多重化およびチャネル・インターリーブの既知の構成が示されている。この構成は、主流のサービスすなわちQoS(サービス品質)チャネルと呼ばれる複数のデータ信号ストリームをまとめて多重化するよう動作するサービス・マルチプレクサ10を備える。複数のデータ信号ストリームはそれぞれのサービスブロック12を介して供給される

が、この図ではそのうちの1つのみが示されている。それぞれのサービスブロック12には、構成要素である複数の入力信号が入力14に供給される。この信号は、たとえば音声、データおよびマルチメディア信号のような様々な種類の信号のうち任意のものを含むことができる。これらの入力信号は、任意の伝送レート、フレームサイズおよび他のパラメータを有することができる。これらの入力信号は、ブロック16においてCRC（巡回冗長検査）コードが付加され、伝送チャネル・マルチプレクサ18においてまとめて多重化される。多重化された信号は、セグメント化ブロック20において符号化のためにセグメント分割され、該セグメント化された信号は、FECブロック22においてFEC（前方誤り訂正）符号化を受ける。符号化された信号は、マルチプレクサ24において多重化される。

【0022】多重化された信号は、ブロック26においてレートマッチング（冗長なデータシンボル（ビット）のバンクチャリング（削除）またはデータシンボル（ビット）の反復）を受け、データレートを、10msの持続時間の無線フレームを持つ無線通信レート（エアレート（air rate））に整合させる。主に隣接したビットを分離して、無線チャネルにおけるフェージングに起因したエラーの悪影響を低減させるために、データビットは、第1のインターリーブ28においてインターリーブされる。第1のインターリーブ28は、それぞれが10msの、データビットのブロックを並べ替えるよう動作するので、チャネル・インターリーバまたはフレーム間インターリーバと呼ばれる。図1では、インターリーブ28はレートマッチングのブロック26に続くものとして示されているが、以下に詳細に説明するようにこれらの機能の配置を交換することができる。その場合、マルチプレクサ24からの多重化された信号はチャネル・インターリーバ28に供給され、インターリーバからのインターリーブされた信号はレートマッチングブロック26に供給される。また、たとえばこれらの機能を、中央局からのダウンリンクの信号伝送のために図1に示される順序にすることができ、または中央局へのアップリンクの信号伝送のために逆の順序にすることもできる。

【0023】機能26および28に続き、結果としてのレートマッチングされインターリーブされた信号は、セグメント化ブロック30および32において、無線フレームおよび物理チャネルについてそれぞれセグメント分割され、マルチプレクサ10により多重化するための信号を生成する。マルチプレクサ10により出力された信号は、第2のインターリーブ34によってインターリーブされる。第2のインターリーブ34の出力は、既知の方法におけるCDMA無線通信経路を介した通信のために、セグメント化およびマッピングブロック36において、セグメント化されて専用の物理チャネルにマップされる。

【0024】上記参照したWen Tong他による特許出願に記載されているように、第1のインターリーブ28は、たとえば以下に説明するような簡単なシャフリング操作にまで、第2のインターリーブ34を省略または縮小することを可能にするのに十分良いパフォーマンスを持つことができる。このことは、以下の理由により特に望ましい。すなわち、そうでなければ第2のインターリーブ34は、それぞれの第1のインターリーブ28によって実行されるインターリービングを劣化させる可能性があるのに対し、それぞれの第1のインターリーブ28を、その特定のレートマッチングされたデータストリームおよびQoSについて最適化することができるからである。

【0025】したがって、第1のインターリーブ28は、十分ランダムな拡散特性を提供する代数的インターリーブとして実現される。それぞれのQoSチャネルについて複数の符号化されたビットブロックまたはデータ伝送フレームは、2次元マトリクスにマップされ、マトリクスの行および列を並べ替えるために線形合同数測（linear congruential rule）を受け、インターリービング機能を実現する。最良のパラメータ・セットを探索することにより、最大のインターリービングの深さおよび期間（タイムスパン（time span））を求めることができる。したがって、インターリーブは、たとえばルックアップテーブルのために大きいサイズのメモリを必要としたり、レートマッチング機能に対応するのが不十分だったりといった、既知のインターリーブの不利な点を持つことなく、比較的簡単な形態を持つことができる。

【0026】以下の説明は、マトリクスの行および列を参照するけれども、これは便宜さおよび明瞭さのためであり、インターリーブの機能を変更することなく行および列を交換することができるということ、実際におよび以下に説明するように、インターリーブは、メモリに格納されたビットをメモリ位置の間で実際に移動することなく、データビットが格納された線形メモリのメモリ位置を読み出しまたは書き込みアドレッシングを実行するといった等価制御により動作することができるということを理解されたい。

【0027】上記参照した特許出願で記述されているように、インターリーブ26は以下のステップを実行するよう動作する。

【0028】1. データビットの符号化されたブロック数が $N_c$ で、それぞれの符号化されたブロックが $N_r$ 個のデータビットの長さであるのを、 $N_r$ 行および $N_c$ 列のマトリクスとして表す。

【0029】2. 以下の式に従ってマトリクスの行および列を並べ替える。

【0030】

【数5】

50 行の並べ替え  $I_r(k) = [\alpha_r k + f_c(1)] \bmod N_r$

列の並べ替え  $I_c(l) = [\alpha_c l + f_r(k)] \bmod N_c$

【0031】ここで、 $I_r(k)$ は行インデックス $k$ のデータビットを表し、 $k$ は $1 \sim N_r$ の整数である。 $\alpha_r$ は整数であり、行並べ替えパラメータである。 $f_c(l)$ は列インデックス $l$ の正関数であり、 $l$ は $1 \sim N_c$ までの整数である。 $I_c(l)$ は列インデックス $l$ のデータビットを表す。 $\alpha_c$ は整数であり、列並べ替えパラメータである。 $f_r(k)$ は行インデックス $k$ の正関数である。 $\bmod N_r$ および $\bmod N_c$ は、モジュロ $N_r$ およびモジュロ $N_c$ 算術をそれぞれ表す。

【0032】3. マトリクスからインターリーブされたデータビットを列ごとに導き出す。

【0033】ステップ1をわずかに変更して、マトリクスの所与の列数をもつ異なる数のデータ伝送フレームに対応するようにすることができる。たとえば、マトリクス\*

$$\begin{bmatrix} N_r / \log_2(\log_2(N_r)) \\ N_c \end{bmatrix}$$

ここで、記号  $\lfloor \cdot \rfloor$  は下の整数に丸める(端数の切捨て)ことを意味する。

【0036】関数  $f_c(l) = m l + [N + 1] \bmod 2$  であり、ここで $m$ は以下の式(5)によって表される数に等しい整数である。また、 $f_r(k) = 2k + [N_r + 1] \bmod 2$  である。

【0037】

$$\lceil N_r / N_c \rceil$$

ここで、記号  $\lceil \cdot \rceil$  は上の整数に丸める(端数の切上げ)ことを意味する。

【0038】 $[N_r + 1] \bmod 2$ は、 $N_r$ が奇数であるときはゼロ、 $N_r$ が偶数であるときは1であることは明らかである。また、 $[N_c + 1] \bmod 2$ は、 $N_c$ が奇数であるときは1、 $N_c$ が偶数であるときはゼロであることは明らかである。よって、関数 $f_c(l)$ および $f_r(k)$ のこれらの部分は、それぞれの数 $N_r$ または $N_c$ が偶数であるとき、簡単な1の加算となる。

【0039】上記に示したように、レートマッチングは、データ伝送フレームのサイズが無線フレームのサイズより大きければ、FEC符号化ブロック22の結果として現れる冗長なデータビットを、最大のパンクチャリング率が伝送フレームサイズの20%になるようパンクチャリング(削除)する。逆に、データ伝送フレームのサイズが無線フレームのサイズより小さければ、伝送フレームのビットは、レートマッチングを達成するよう反復される。レートマッチングは、可能な限り、パンクチャリングされるビットとビット間の分離距離を最大にして、それぞれの無線フレームにおけるパンクチャリングされたビット数を一様にする、すなわち最大の分離を持つ無線フレームの間でパンクチャリングされたビットを均一に分散させることが望ましい。

【0040】図1に示されるようにレートマッチングブロック26がチャンネル・インターリーバ28に先行している場合、図2に示されるような既知のレートマッ

\*スは、 $N_c / \gamma$ 個のデータ伝送フレームについて $N_c = 8$ の列を持つことができ(ここで $\gamma = 1, 2, 4$ または8である)、それに応じて、マトリクスは $N_r / \gamma$ 個の行を持ち、結果として起こるステップ3の変更は、1無線フレームあたりマトリクスの $\gamma$ 個の列を読み出すことである。以下の説明では簡単にするため、 $N_c = 8$ で $\gamma = 1$ と仮定する。

【0034】ステップ2では、行並べ替えパラメータ $\alpha_r$ は、以下の式(3)で表される数より小さい最大素数であるよう選択される。列並べ替えパラメータ $\alpha_c$ は、以下の式(4)で表される数より小さい最大素数であるよう選択される。

【0035】

【数6】

式(3)

式(4)

ング方法を使用することができる。

【0041】図2を参照すると、セグメントサイズ $N_i$ ビットのそれぞれの無線フレームについて、ブロック40において整数 $y$ が $y = N_r - N_i$ のように求められ、 $y$ は、パンクチャリングが必要とされる場合にはゼロより大きく(正)、 $|y|$ ビットの反復が必要とされる場合にはゼロより小さく(負)なる。また、パンクチャリングも反復も必要とされないとき、 $y$ はゼロとなり、この場合ストップ・ブロック41に達する。図2に示されるステップは、 $y$ およびパンクチャリングを使用する代わりに $|y|$ および反復を使用することをのぞき、ビット反復( $y < 0$ 、図2の右側に示される)のステップは、図2の左側に示されるパンクチャリング( $y > 0$ )のステップと実質的に同じである。よって、パンクチャリングの場合についてのみ以下に詳細に説明する。

【0042】 $y > 0$ ならば、伝送フレームの $N_r$ ビットのうち $y$ のパンクチャリングが、無線フレームの $N_i$ ビットを生成するのに必要とされる。この場合、ブロック42において、パラメータ $e$ が、特定の無線フレームについての所望の任意の方法で決定される開始オフセット $e_0$ に初期化され、行カウンタ $r$ は1に初期化される。ブロック43において、 $r \leq N_r$ かどうか判断され、 $r \leq N_r$ ならば、ブロック44において $e$ の値が2 $y$ だけ減らされる。次の判断ブロック45において、 $e$

$\leq 0$ かどうか判断され、 $e \leq 0$ ならば、行  $r$  のビットがブロック 46 においてバンクチャリングされ、ブロック 47 において  $e$  の値が  $2N_r$  だけ増やされ、行カウンタ  $r$  はブロック 48 において 1 だけ増やされ、判断ブロック 43 に戻る。ブロック 45 において  $e \leq 0$  でなければ（すなわち、 $e > 0$ ）、バンクチャリングを行うことも  $e$  の値を変更することなく、行カウンタ  $r$  をインクリメントするブロック 48 を介してブロック 43 に戻る。ブロック 43 において  $r \leq N_r$  でなければ（すなわち、 $r > N_r$ ）、これはフレームの終わりに達したことを示すので、ストップ・ブロック 41 においてシーケンスを終了させる。

【0043】しかしながら、レートマッチングブロック 26 がチャンネル・インターリーバ 28 に続く場合には、レートマッチングは、並べ替えられた（インターリーブされた）ビットストリームに対して実行されるので、レートマッチングの問題が非常に複雑になる。一般に、チャンネル・インターリーピング処理の要件とレートマッチング処理の要件は両立しない。

【0044】チャンネル・インターリーピング処理後のビットのマトリクス内におけるバンクチャリングまたは反復されるビットの、適切で望ましく最適化されたレートマッチングパターンの設計は、かなりな複雑さ、または非現実的な作業を意味する。この発明は、インターリーピング前のマトリクスについて、バンクチャリングまたは反復されるビットの適切で望ましく最適化されたレートマッチングパターンを提供し、チャンネル・インターリーバの出力においてバンクチャリングされるべきまたは反復されるべき対応するビットを、デインターリーピングすなわち復号化処理を使用して判断することにより、この問題を回避する。この処理は、以下にさらに記述するように、インターリーピング処理と全く同じ構造によってデインターリーピングすなわち復号化処理を実現することができるという事実により、容易化される。便宜上および明瞭さのため、以下の説明は、インターリーピング前（またはデインターリーピング後）のビットのマトリクスを、ナチュラル・マトリクス  $NM$  と呼び、インターリーピング後のビットのマトリクスをランダム化マトリクス  $RM$  と呼ぶ。

【0045】図 3 は、この発明の実施形態に従うチャンネル・インターリーバ 28 およびレートマッチングブロック 26 の実現を示す。図 3 に示されるように、インターリーバ 28 は、2 等分された作業用メモリ 50 を備え、これは、該メモリに書き込み、該メモリから読み出す既知の方法で交互に使用され、それぞれ上記に説明したマトリクスで表される  $N_r N_c$  のデータビットを格納する。これらのデータビットは、マトリクスの行単位の編成に線形的に対応するメモリに書き込まれる。モジュロ  $N_r$  行カウンタ 51 は、クロック信号  $CLK$  に応答して、行インデックス  $k$  を表すカウントを提供し、このカ

ウンタ 51 のキャリ（桁上げ）出力は、列インデックス  $l$  を表すカウントを提供するモジュロ  $N_c$  の列カウンタ 52 に供給される。カウンタ 51 および 52 のカウンタ  $k$  および  $l$  は、図 3 の破線内に示されるアドレス符号器に供給される。より具体的には、列カウンタ 53 のカウンタは乗算器 54 および 55 に供給され、乗算器 54 および 55 にはまた、パラメータ  $\alpha_c$  および  $m$  がそれぞれ供給され、 $\alpha_c l$  および  $m l$  を表す積をそれぞれ生成する。行カウンタ 51 のカウンタは乗算器 56 および 57 に供給され、乗算器 56 および 57 にはまた、整数 2 およびパラメータ  $\alpha_r$  がそれぞれ供給され、 $2k$  および  $\alpha_r k$  を表す積をそれぞれ生成する。加算器 58 は、乗算器 54 および 56 の出力を加算し、 $N_c$  が偶数か奇数かに依存してそれぞれ 1 または 0 を選択的に加算する。加算器 58 の出力は、モジュロ機能 59 によってモジュロ  $N_c$  の形式になり、上記に説明した列並べ替え機能を完了する。加算器 60 は乗算器 55 および 57 の出力を加算し、 $N_r$  が偶数か奇数かに依存してそれぞれ 1 または 0 を選択的に加算する。加算器 60 の出力は、モジュロ機能 61 によってモジュロ  $N_r$  の形式になり、上記説明した行並べ替え機能を完了させる。モジュロ機能 59 および 61 のそれぞれは、比較機能および減算機能を備えることができる。機能 59 および 61 の出力は、読み出しアドレス結合器により組み合わせられ、データビットのインターリーブされるシーケンスにおけるそれぞれのデータビットをメモリ 50 から読み出すためのアドレスを生成する。図 3 に示されるように、読み出しアドレスは、以下に説明するように提供されるスイッチ 63 を介してメモリ 50 に供給される。

【0046】行  $N_r$  の数が 2 の累乗であるならば、アドレス結合器 62 は、メモリ 50 の読み出しアドレスの、下位ビットとしてモジュロ機能 61 の出力と、上位ビットとしてモジュロ機能 59 の出力とを単に組み合わせることができる。これは、アドレス結合器 62 により、機能 61 の出力が機能 59 の出力に  $N_r$  回加算されることに相当する。

【0047】 $N_c$  の整数倍ではない任意のサイズのフレームのデータビットをインターリーブするのが望ましい場合がある。この場合、マトリクスの行数は、インターリーブされるべきデータビットのすべてを収容するよう選択され、作業用メモリ 50 の最後の数個（ $N_c$  より少ない）のメモリ位置は書き込まれていない。インターリーブされるデータビットからこれらのメモリ位置のデータビットを切り捨てるため、図 3 のインターリーバ 28 はまた、アドレス結合器 62 の読み出しアドレス出力に、これらのメモリ位置を検出する復号器 64 を備え、このようなメモリ位置が検出されると、これらの位置についてはメモリ 50 からデータが読み出されることを防ぐよう、スイッチ 63 を開く。メモリ 50 からインターリーブされたデータビットの一定のデータ出力レートを

提供するため、さらに図3のインターリーブ28はFIFO（先入れ先出し）メモリ65を備え、FIFOメモリ65はクロック信号CLKによってクロックされる。このクロック信号CLKを介して、インターリーブされたデータビットはインターリーブの出力ライン66に供給される。FIFO65は、それぞれのインターリーブ処理の開始時に予め満たされ、読み出されない、よって切り捨てられるメモリ位置を許容するのに十分なサイズ（たとえば、最大 $N_c$ ）を持つ。

【0048】ライン66上のインターリーブされたデータビットは、図3に示されるレートマッチングブロックすなわちレートマッチング機能26に供給される。このレートマッチング機能は、クロック信号が供給されるレートマッチング・アドレス生成器70と、アドレス・セパレータ71、アドレス復号器72、バッファすなわち記憶部73、比較器74およびライン76上にレートマッチングされたデータ出力を提供するデータビット選択器75を備える。インターリーブ28の一定のデータビット・レート出力を提供するFIFO65の備えと同様に、レートマッチング機能26も、FIFOまたは他のバッファ（図示せず）を有して、出力ライン76から一定レートのデータビットを提供することができる。

【0049】レートマッチング・アドレス生成器70は、その出力において、以下にさらに説明するように、この処理について求められるバンクチャリングまたは反復パターンに従う、それぞれのバンクチャリングされるビットまたは反復されるビットのナチュラル・マトリクスアドレスを生成する。このアドレスは、アドレス・セパレータ71によって上位要素および下位要素に分離されるが、この処理は、上記説明した読み出しアドレス結合器62のものとは反対である。こうして、行数 $N_r$ が2の累乗ならば、アドレス・セパレータ81は、生成器70によってアドレスビットの出力を簡単に上位ビットおよび下位ビットに分離することができる。このことは、生成器70からのアドレスを $N_r$ で除算して、整数の商および余りを生成することに相当する。この商および余りが、アドレス・セパレータ71の2つの出力を構成する。

【0050】アドレス復号器72は、アドレス符号器53とは逆の機能を実行する。上記に示したように、ここで説明する代数的インターリーブ処理ならば、ディンターリーバーの構造を、インターリーブの構造と全く同じものとすることができ、それに応じてアドレス復号器72を、アドレス符号器53と全く同じものにすることができる。したがって、アドレス復号器72の詳細な構造については図3に図示されていないが、これは、図3に示されるアドレス符号器の構造と同一である。インターリーブおよびディンターリーブの相補的な処理のこの同一構造特性は、これらの機能を実現する際に重要な有利さおよび簡略さを与える、ということは明

らかであろう。

【0051】アドレス復号器72の出力は記憶部73にバッファされ、記憶部73からのバッファされた出力は、比較器74において、チャンネル・インターリーブ28の行カウンタ51および列カウンタ52の現在のカウンタ $k$ および $l$ とそれぞれ比較され、比較された値が同じであるときは、ライン77上に予め決められた状態の選択器制御信号を提供する。このように、選択器制御信号は、ライン66上のビットがバンクチャリングされるか反復されるかするたびごとに、その状態でライン77上に生成される。そのほかの時は、バンクチャリングも反復もされないビットについて、ライン77上の制御信号は、クロック信号CLKに同期して、選択器75の3つの入力の中の真ん中の1つ（図3に示される）に供給されるライン66からのビットを出力ライン76に供給するよう、選択器75を制御する。反復またはバンクチャリングされるべきそれぞれのビットのとき、ライン77上の制御信号は、ビットが反復されるかバンクチャリングされるかにそれぞれ依存して（これは、選択器75への制御入力P/Rによって決定される）、その上側入力またはその下側入力（図3に示される）のどちらかから、ビットをその出力ライン76に供給するよう、選択器75を代わりに制御する。選択器75の上側の入力は出力ライン76に接続されてビット反復を提供し、選択器75の下側の入力は、図に示されるように接続を持たず、ビットのバンクチャリングを提供する。上記に示されるように、インターリーブされたレートマッチングされたデータビットについての一定の出力データビット・レートはバッファ（図示せず）により提供され、そこに、出力ライン76上のデータビットが供給される。

【0052】レートマッチング機能26の復号器72によって提供されるアドレス復号化のために、レートマッチング・アドレス生成器70は、図2を参照して上記説明したやり方で、このパターンを最適化する所望の方法で求められた1つのパラメータ $e_{0..s}$ を使用して、バンクチャリングまたは反復されるビットの所望のパターンを通常のマトリクスのアドレスによって簡単に求めることができる。たとえば、このパラメータを、 $e_{0..s} = [2py + 1] \bmod 2N_r$  のような式によって求めることができ、ここで上記説明したように、 $y$ は、マトリクスのそれぞれの列についてバンクチャリングまたは反復されるビット数であり、 $p$ は0～7の列インデックスである（ $N_c = 8$ の場合について）。

【0053】それぞれが100ビットの8個のデータ伝送フレームの上記説明したインターリーブで、それぞれ8ビットのチャンネルインターリーブされレートマッチングされた無線フレームを生成するのに最大20%のバンクチャリング率を必要とする例を、以下の表1、2および3で示す。 $N_c = 8$ および $N_r = 10$ である。表1は、 $10 \times 8$ のナチュラル・マトリクスの行ごとに、



0～79まで番号付けされた80個のデータビットのエントリを示し、行インデックスkは1～10であり、列インデックスlは1～8である。

\*【0054】

【表1】

\*

		l							
		1	2	3	4	5	6	7	8
k	1	0	1	2	3	4	5	6	7
	2	8	9	10	11	12	13	14	15
	3	16	17	18	19	20	21	22	23
	4	24	25	26	27	28	29	30	31
	5	32	33	34	35	36	37	38	39
	6	40	41	42	43	44	45	46	47
	7	48	49	50	51	52	53	54	55
	8	56	57	58	59	60	61	62	63
	9	64	65	66	67	68	69	70	71
	10	72	73	74	75	76	77	78	79

【0055】上記説明したチャネル・インターリーブングは、以下の表2によって示されるランダム化されたマトリクスを生成する。

※【0056】

【表2】

※

		l							
		1	2	3	4	5	6	7	8
k	1	57	40	79	62	45	28	11	74
	2	35	18	1	64	23	6	69	52
	3	13	76	59	42	25	8	47	30
	4	71	54	37	20	3	66	49	32
	5	73	56	15	78	61	44	27	10
	6	51	34	17	0	39	22	5	68
	7	29	12	75	58	41	24	63	46
	8	7	70	53	36	19	2	65	48
	9	9	72	31	14	77	60	43	26
	10	67	50	33	16	55	38	21	4

【0057】その後、上記説明したレートマッチングは、レートマッチングのアルゴリズムによって生成されたパターン16ビットをバンクチャリングする。すなわち、ランダム化されたマトリクスのそれぞれの列から2ビットをバンクチャリングし、以下の表3に示される★

★バンクチャリングされたランダム化マトリクスを提供する。

【0058】

【表3】

		l							
		1	2	3	4	5	6	7	8
k	1	57	40	79	62	45	28		74
	2	35	18	1		23	6	69	52
	3	13	76	59	42		8		30
	4	71		37	20	3	66	49	
	5	73	56	15	78		44	27	10
	6	51		17	0	39	22	5	
	7		12		58	41	24	63	46
	8	7	70	53	36	19		65	48
	9		72		14	77	60	43	26
	10	67	50	33		55		21	4

【0059】チャネル・インターリーブされたレートマッチングされたデータビットは、表3から列ごとに導き出される。すなわち、[57, 35, ..., 51, 7, 67, 40, ..., 26, 4]の順番に導き出される。バンクチャリングされたビットは、2, 9, 11, 16, 25, 29, 31, 32, 34, 38, 47, 54, 61, 64, 68および75であり、それらの最大バンクチャ距離は9(25-16より)であり、最小バンクチャ距離は1(32-31より)である。この小さい最小バンクチャ距離は、この特定の例が最適ではないことを示し、より大きい最小バンクチャ距離が望ましい。パラメータの非常に多くの他の決定、特にパラメータe<sub>0</sub>の決定を、バンクチャリング処理を最適化するように提供することができる。

【0060】上記に示したように、第2のインターリーブ34の処理は、第1のインターリーブ28の結果とし

て達成されるパフォーマンスを劣化させないことが望ましく、このため、第2のインターリーブ34は、簡単なシャフリング処理にまで縮小され、これにより、第2のインターリーブ34は、それぞれのQoSデータストリームについて第1のインターリーブ28によって達成される拡散特性を維持しつつ、異なるQoSを持つデータストリームをインターリーブする。

【0061】図4は、図1のサービス・マルチプレクサ10を介して提供されるそれぞれのサービスブロック12から、上記説明したように提供されるインターリーブされた無線フレームの2つのデータストリームのビットをインターリーブするのに有利に使用することのできるビット・シャフリング・アルゴリズムのフローチャートを示す。N<sub>1</sub>ビットのフレームを持つ1つのストリームTQ<sub>1</sub>およびN<sub>2</sub>ビットのフレームを持つ第2のストリ

ームTQ<sub>2</sub>を示し、N<sub>1</sub> ≥ N<sub>2</sub>である。図4は、ストリームTQ<sub>2</sub>のビットが、ストリームTQ<sub>1</sub>にどのようにして挿入されるかを示す。

【0062】図4を参照すると、最初にブロック82において、パラメータeがN<sub>1</sub>に、カウンタrが1に初期化される。ブロック83において、r ≤ N<sub>1</sub>かどうか判断され、r ≤ N<sub>1</sub>ならば、ブロック84においてeの値が2N<sub>2</sub>だけ減らされる。次の判断ブロック85においてe ≤ 0かどうか判断され、e ≤ 0ならば、ブロック86において、ストリームTQ<sub>2</sub>の次のビットがストリームTQ<sub>1</sub>に挿入される。eの値は、ブロック87において2N<sub>1</sub>だけ増やされ、カウンタrはブロック88において1だけ増やされ、その後判断ブロック83に戻る。ブロック85においてe ≤ 0でなければ(すなわち、e > 0)、任意のビット挿入もeの値変更も行うことなく、カウンタrをインクリメントするブロック88を介してブロック83に戻る。ブロック83においてr ≤ N<sub>1</sub>でなければ(すなわち、r > N<sub>1</sub>)、これはフレームの終わりに達したことを示すので、ストップ・ブロック81でシーケンスを終える。

【0063】2より多くのデータストリームについては、連続するデータストリームについて同じ処理が帰納的に適用される。上記の説明および図4から、この処理のステップが、図2のバンクチャリングおよび反復処理のステップと直接的な相関関係を持ち、よってこの帰納的なシャフリング処理の実行が特に便宜であるということとは明らかであろう。

【0064】上記に説明したように、所望のレートマッチングを達成するためのビットのバンクチャリングは、符号器22によって提供されるFEC符号化に起因した冗長性を持つデータビットに適用される。符号化の1つの好ましい形は、いわゆるターボ(並列接続畳み込み)符号化である。この場合、符号化されたデータビットは入力データビットそのものを有し、これを組織(systematic)データビットSと呼ぶ。さらに、符号化されたデータビットはパリティビットP1およびP2を有し、これらは、入力データビットおよびインターリーブされた入力データビットについて処理する畳み込み符号器によって提供される。パリティビットP1およびP2は、典型的にはターボ符号器内でバンクチャリングされ、所望のレートのターボ符号器を提供する。ターボ符号器によって構成される符号器22については、次のレートマッチング機能26が、組織ビットSのいずれをもバンクチャリングしないが、パリティビットP1および(または)P2のみをバンクチャリングすることを確実にする必要がある。反復の場合には、組織ビットSの反復の2倍または3倍のオーダーを持つファクターによって、パリティビットP1およびP2を反復することにより、パフォーマンス利得を提供することが求められる。

【0065】このため、図5は、ターボ符号化によって

符号化されたデータのチャンネル・インターリーブングおよびレートマッチングについての、図1の構成の一部に修正を加えたものを示す。図5を参照すると、FEC符号器22のうちの1つを構成するターボ符号器は、破線ボックス90内に示され、周知のように、入力データビットをインターリーブするターボ符号インターリーバ91と、インターリーブングの前後で入力データビットを処理し、パリティビットP1およびP2を生成する2つの従来の符号器92を備える。また、入力データビットは、組織ビットSとして符号器出力に供給される。また、図示しないバンクチャリング・ブロックを、符号器出力への供給のために、パリティビットP1およびP2のいくつかのみを選択して、符号器出力に供給させるために設けることができる。

【0066】上記説明した1つのチャンネル・インターリーバの代わりに、図5は、組織ビットストリームおよびパリティビットストリームに、個々のチャンネル・インターリーバ93が提供されることを示す。図5に示されるように、3つのチャンネル・インターリーバ93があるが、パリティビットP1およびP2のストリームを組み合わせる一緒にインターリーブするようにして、2つのみのチャンネル・インターリーブを設け、一方は組織ビットストリーム用に、他方はパリティビットストリーム用にすることができるのは、明らかである。図5のチャンネル・インターリーバ93へのさらなる入力は、複数のチャンネルについての組織ビットストリームおよびパリティビットストリームの多重化をそれぞれ示し、これは図1のマルチプレクサ24に対応するものである。

【0067】チャンネル・インターリーバ93に続くレートマッチング機能を、破線ボックス94内に示す。バンクチャリング機能95を、チャンネル・インターリーブされたパリティビットストリームにのみ適用し、それに対し反復機能96を、パリティビットストリームおよび組織ビットストリームに提供することができる。それに応じて、選択器97が、チャンネル・インターリーブされたビットを結合するよう示されている。バンクチャリングおよび反復は、上記に説明したものと同じであることができる。この点で、図5に示されるものが、レートマッチング機能の実際の実現を示すというよりも、バンクチャリングが組織ビットに適用されない原則を図によって表すことを意図したものである、ということは明らかであろう。たとえば、必要とされるバンクチャリングまたは反復を、パリティビットストリームにのみ適用して、組織ビットストリームのバンクチャリングも反復も行うことなく、所望のレートマッチングを提供することもできるということは明らかであろう。

【0068】上記の説明は、ここで記述される様々な処理について分離した機能およびユニットについて参照したけれども、1または複数のデジタル信号プロセッサまたは他の集積回路の機能を使用して、多くの形態を実現

することができるということは明らかであろう。

【0069】この発明の特定の実施形態を説明してきたが、多くの改良、変更および適合を、この発明の範囲から離れることなく行うことができるということは明らかであろう。

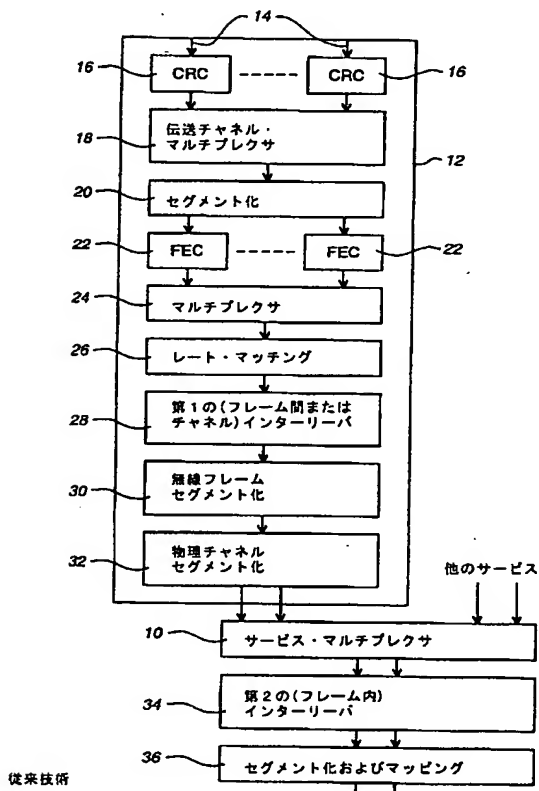
【図面の簡単な説明】

【図1】第3世代CDMA通信システムにおけるサービス多重化およびチャネル・インターリーピングのための既知の構成を示す図。

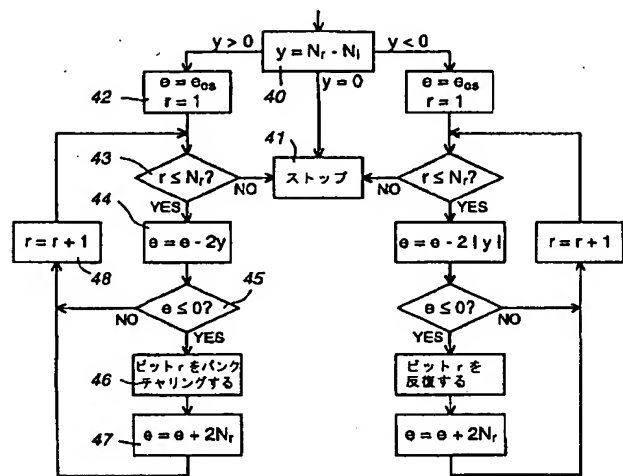
【図2】既知のレートマッチングアルゴリズムに関する

10 26 レートマッチング  
28 インターリーバ

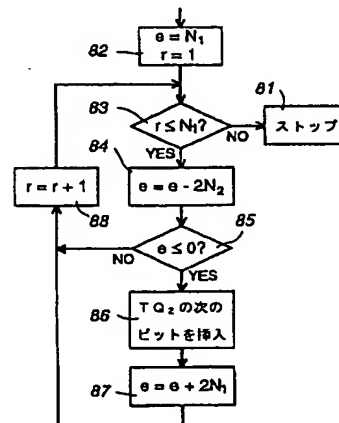
【図1】



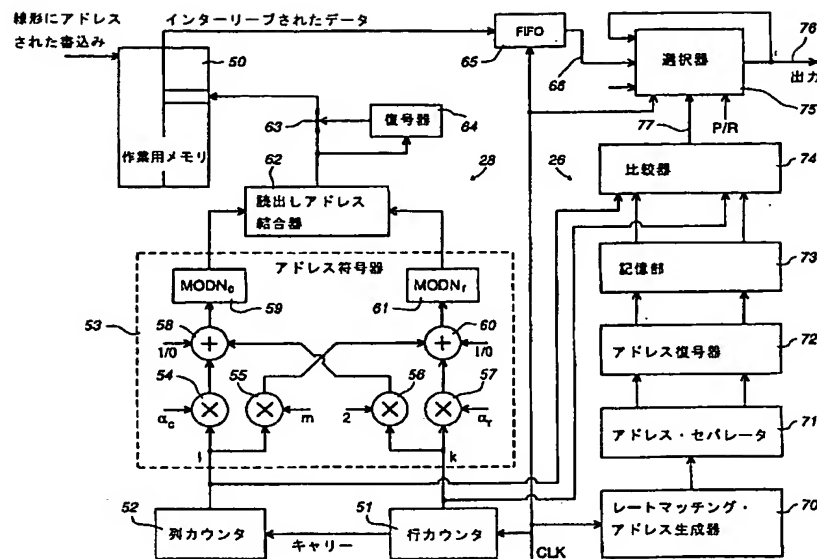
【図2】



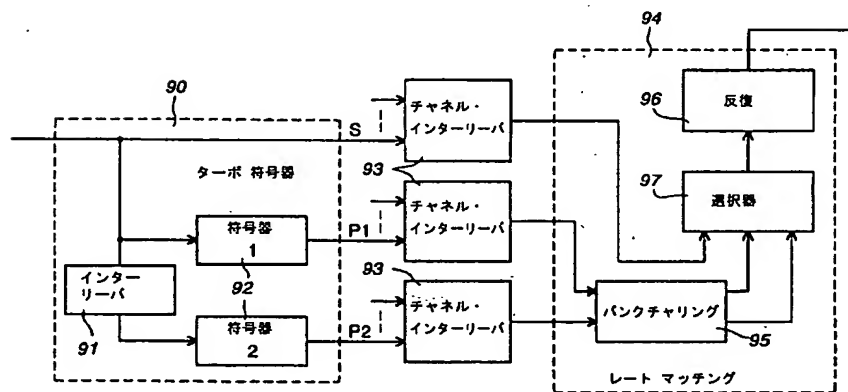
【図4】



【図 3】



【図 5】



フロントページの続き

(72)発明者 ウェン・トン  
カナダ、ケー2シー、3エル7、オンタリ  
オ、オタワ、キャッスルヒル・クレッセン  
ト、ナンバー 903-1000

(72)発明者 キャサリン・ゴースー  
フランス75015パリ、ル・デ・ヴォージラ  
ール 147  
(72)発明者 ステファン・ゴスン  
フランス92310イシイー・レ・ムリネーク  
ス、ル・エドワード・ブランリー 6

【外国語明細書】

1. Title of Invention

**RATE MATCHING AND CHANNEL INTERLEAVING  
FOR A COMMUNICATIONS SYSTEM**

2. Claims

1. A method of matching a rate of data bits, in a matrix of data bits interleaved by a predetermined interleaving process, to a desired rate by deletion of redundant data bits or repetition of data bits derived from the matrix, including the steps of:
  - determining in a non-interleaved matrix of said data bits a pattern of bits to be deleted or repeated to provide said desired data rate;
  - decoding an address of each bit in said pattern in a manner inverse to the interleaving process to produce a respective address of the bit in the matrix of interleaved data bits; and
  - deleting or repeating the respective bit in the interleaved data bits in dependence upon the respective address.
2. A method as claimed in claim 1 wherein the address decoding is performed in the same manner as a coding of addresses for producing the interleaved data bits from the non-interleaved matrix of said data bits.
3. A method as claimed in claim 1 or 2 wherein the pattern of bits to be deleted or repeated is dependent upon the number of bits to be deleted or repeated and a column index of the matrix.
4. A method as claimed in claim 1, 2, or 3 wherein the interleaving process comprises permuting rows and columns of a matrix of  $N_r$  rows and  $N_c$  columns, in which data bits to be interleaved are represented row by row, in accordance with:
  - Row Permutation  $I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$
  - Column Permutation  $I_c(l) = [\alpha_c l + f_r(k)] \bmod N_c$
 where  $I_r(k)$  represents a data bit with a row index  $k$ ,  $k$  is an integer from 1 to  $N_r$ ,  $\alpha_r$  is an integer,  $f_c(l)$  is a non-zero function of a column index  $l$ ,  $l$  is an integer from 1 to  $N_c$ ,  $I_c(l)$  represents a data bit with the column index  $l$ ,  $\alpha_c$  is an integer,  $f_r(k)$  is zero or a function of the row index  $k$ , and  $\bmod N_r$  and  $\bmod N_c$  represent modulo- $N_r$  and modulo- $N_c$  arithmetic respectively, interleaved data bits being derived from the matrix column by column.
5. A method as claimed in claim 4 wherein  $f_c(l) = ml + [N_r + 1] \bmod 2$ , where  $m$  is an integer.
6. A method as claimed in claim 5 wherein  $m$  is approximately equal to  $N_r / N_c$ .
7. A method as claimed in claim 4, 5 or 6 wherein  $f_r(k) = 2k + [N_c + 1] \bmod 2$ .
8. A method as claimed in any of claims 4 to 7 wherein  $\alpha_r$  is the largest prime number less than  $N_r / \log_2(\log_2(N_r))$ .

9. Rate matching apparatus arranged for carrying out a method as claimed any of claims 1 to 8.

10. A method of interleaving data bits comprising permuting rows and columns of a matrix of  $N_r$  rows and  $N_c$  columns, in which data bits to be interleaved are represented row by row, in accordance with:

$$\text{Row Permutation } I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$$

$$\text{Column Permutation } I_c(l) = [\alpha_c l + f_r(k)] \bmod N_c$$

where  $I_r(k)$  represents a data bit with a row index  $k$ ,  $k$  is an integer from 1 to  $N_r$ ,  $\alpha_r$  is an integer,  $f_c(l) = ml + [N_r + 1] \bmod 2$  is a non-zero function of a column index  $l$ ,  $l$  is an integer from 1 to  $N_c$ ,  $m$  is an integer,  $I_c(l)$  represents a data bit with the column index  $l$ ,  $\alpha_c$  is an integer,  $f_r(k) = 2k + [N_c + 1] \bmod 2$ , and  $\bmod 2$ ,  $\bmod N_r$  and  $\bmod N_c$  represent modulo-2, modulo- $N_r$ , and modulo- $N_c$  arithmetic respectively, interleaved data bits being derived from the matrix column by column.

11. A method as claimed in claim 10 wherein  $m$  is approximately equal to  $N_r / N_c$ .

12. A method as claimed in any of claim 10 or 11 wherein  $\alpha_r$  is the largest prime number less than  $N_r / \log_2(\log_2(N_r))$ .

13. A data interleaver arranged for carrying out the method of claim 10, 11, or 12.

14. A method of interleaving and rate matching parallel concatenated convolutional coded data by deletion of coded data bits, the coded data bits comprising systematic bits and parity bits, including the steps of interleaving the systematic bits separately from the parity bits, and deleting parity bits from the interleaved parity bits to provide the rate matching.

15. A method of interleaving and rate matching parallel concatenated convolutional coded data by repetition of coded data bits, the coded data bits comprising systematic bits and parity bits, including the steps of interleaving the systematic bits separately from the parity bits, and repeating parity bits of the interleaved parity bits with a greater repetition factor than any repetition of systematic bits of the interleaved systematic bits, to provide the rate matching.

16. Coding, interleaving, and rate matching apparatus arranged to carry out the method of claim 14 or 15.

17. A method or apparatus generally, substantially, or essentially as described herein with reference to any one or more of Figs. 3 to 5 of the accompanying drawings.

### 3. Detailed Description of Invention

This invention relates to rate matching and channel interleaving for a communications system.

#### Background of the Invention

It is well known to perform interleaving of data in a communications system using forward error correction (FEC) in order, on deinterleaving, to distribute errors to facilitate their correction. Typically, such interleaving uses a block interleaver to interleave blocks of data. So-called turbo coding (parallel concatenated convolutional coding) uses an interleaver between inputs to two convolutional coders which produce respective parity bits from the input data before and after interleaving. With increasing attention being given to the use of turbo coding, particularly in wireless communications systems, attention has also been given to the form of the interleaver.

So-called 3rd generation CDMA (code division multiple access) wireless communications systems are also being developed which require a channel or inter-frame interleaver which operates to interleave or permute data in blocks corresponding to the radio frame duration, typically 10 ms. In such systems the channel interleaver either precedes or follows a rate matching function which serves to match various data rates to the radio frame rate, and which typically involves puncturing (omission) or repetition of data symbols, in this case data bits. It is desirable to distribute the omitted or repeated bits as evenly as possible, with as great a distance as possible between punctured or repeated bits in the de-interleaved frames, in a manner that is easy to implement and that is relatively independent of variables such as the frame size, number of frames, and puncturing rate.

Canadian patent application No. (File 10378RO) filed March 19, 1999 in the names of Wen Tong et al., entitled "Data Interleaver And Method Of Interleaving Data", describes and claims a method of interleaving data and a data interleaver which advantageously can be used to provide the channel interleaving referred to above. The present invention is concerned with rate matching in a manner which can be used with particular advantage for data after such channel interleaving, but which is also applicable to other forms of interleaved data. This invention also provides improvements in and further applications of such channel interleaving.

#### Summary of the Invention

According to one aspect, this invention provides a method of matching a rate of data bits, in a matrix of data bits interleaved by a predetermined interleaving process, to a desired rate by deletion of redundant data bits or repetition of data bits derived from the matrix, including the steps of: determining in a non-interleaved matrix of said data bits a pattern of bits to be deleted or repeated to provide said desired data rate; decoding an



address of each bit in said pattern in a manner inverse to the interleaving process to produce a respective address of the bit in the matrix of interleaved data bits; and deleting or repeating the respective bit in the interleaved data bits in dependence upon the respective address.

It is particularly advantageous, and may be necessary in practice, for the address decoding to be performed in the same manner as a coding of addresses for producing the interleaved data bits from the non-interleaved matrix of said data bits. This is facilitated in preferred embodiments of the method of the invention by the interleaving process comprising permuting rows and columns of a matrix of  $N_r$  rows and  $N_c$  columns, in which data bits to be interleaved are represented row by row, in accordance with:

$$\text{Row Permutation } I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$$

$$\text{Column Permutation } I_c(l) = [\alpha_c l + f_r(k)] \bmod N_c$$

where  $I_r(k)$  represents a data bit with a row index  $k$ ,  $k$  is an integer from 1 to  $N_r$ ,  $\alpha_r$  is an integer,  $f_c(l)$  is a non-zero function of a column index  $l$ ,  $l$  is an integer from 1 to  $N_c$ ,  $I_c(l)$  represents a data bit with the column index  $l$ ,  $\alpha_c$  is an integer,  $f_r(k)$  is zero or a function of the row index  $k$ , and  $\bmod N_r$  and  $\bmod N_c$  represent modulo- $N_r$  and modulo- $N_c$  arithmetic respectively, interleaved data bits being derived from the matrix column by column.

It is currently considered optimum to choose  $f_c(l) = ml + [N_r + 1] \bmod 2$ , where  $m$  is an integer,  $m$  approximately equal to  $N_r / N_c$ ,  $f_r(k) = 2k + [N_c + 1] \bmod 2$ , and  $\alpha_r$  as the largest prime number less than  $N_r / \log_2(\log_2(N_r))$ .

The invention also provides rate matching apparatus arranged for carrying out a method as recited above.

Another aspect of this invention provides a method of interleaving data bits comprising permuting rows and columns of a matrix of  $N_r$  rows and  $N_c$  columns, in which data bits to be interleaved are represented row by row, in accordance with:

$$\text{Row Permutation } I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$$

$$\text{Column Permutation } I_c(l) = [\alpha_c l + f_r(k)] \bmod N_c$$

where  $I_r(k)$  represents a data bit with a row index  $k$ ,  $k$  is an integer from 1 to  $N_r$ ,  $\alpha_r$  is an integer,  $f_c(l) = ml + [N_r + 1] \bmod 2$  is a non-zero function of a column index  $l$ ,  $l$  is an integer from 1 to  $N_c$ ,  $m$  is an integer,  $I_c(l)$  represents a data bit with the column index  $l$ ,  $\alpha_c$  is an integer,  $f_r(k) = 2k + [N_c + 1] \bmod 2$ , and  $\bmod 2$ ,  $\bmod N_r$  and  $\bmod N_c$  represent modulo-2, modulo- $N_r$ , and modulo- $N_c$  arithmetic respectively, interleaved data bits being derived from the matrix column by column.

The invention also provides a data interleaver arranged for carrying out this method.

Another aspect of the invention provides a method of interleaving and rate matching parallel concatenated convolutional coded data by deletion of coded data bits, the coded data bits comprising systematic bits and parity bits, including the steps of

interleaving the systematic bits separately from the parity bits, and deleting parity bits from the interleaved parity bits to provide the rate matching.

A further aspect of the invention provides a method of interleaving and rate matching parallel concatenated convolutional coded data by repetition of coded data bits, the coded data bits comprising systematic bits and parity bits, including the steps of interleaving the systematic bits separately from the parity bits, and repeating parity bits of the interleaved parity bits with a greater repetition factor than any repetition of systematic bits of the interleaved systematic bits, to provide the rate matching.

The invention further provides coding, interleaving, and rate matching apparatus arranged to carry out these methods.

Yet another aspect of this invention relates to a method of shuffling interleaved and rate matched data streams in the manner described below with reference to Fig. 4 of the drawings, and to the recursive application of this method to more than two such data streams.

Further aspects of the invention relate to a method or apparatus generally, substantially, or essentially as described below with reference to any one or more of Figs. 3 to 5 of the drawings.

#### Detailed Description

Referring to Fig. 1, there is illustrated a known arrangement for service multiplexing and channel interleaving in a 3rd generation CDMA radio communications system. The arrangement includes a service multiplexer 10 which serves to multiplex together a plurality of data signal streams, referred to as main stream services or QoS (Quality of Service) channels, which are supplied via respective service blocks 12 only one of which is illustrated. Each service block 12 is supplied at inputs 14 with a plurality of constituent input signals, which may for example comprise any of various types of

signals such as voice, data, and multi-media signals. These input signals may have arbitrary transmission rates, frame sizes, and other parameters. The input signals have CRC (cyclic redundancy check) codes added in blocks 16 and are multiplexed together in a transport channel multiplexer 18. The multiplexed signals are segmented, for encoding, in a segmentation block 20, and the segmented signals are subjected to FEC (forward error correction) coding in FEC blocks 22. The encoded signals are multiplexed in a multiplexer 24.

The multiplexed signals are subjected to rate matching (puncturing (deletion) of redundant data symbols (bits) or repetition of data symbols (bits)) in a block 26 to match the data rate to the radio communications rate (air rate) with radio frames of 10 ms duration. Primarily in order to separate adjacent bits to reduce the adverse effects of errors due to fading in the radio channel, the data bits are interleaved in a first interleaver 28, which is referred to as a channel or inter-frame interleaver because it operates to permute blocks each of 10 ms of data bits. Although in Fig. 1 the interleaver 28 is shown following the rate matching block 26, as discussed further below the positions of these functions may be interchanged, the multiplexed signals from the multiplexer 24 being supplied to the channel interleaver 28, and the interleaved signals from the interleaver being supplied to the rate matching block 26. For example, these functions may be in the order shown in Fig. 1 for downlink transmission of signals from a central station, and may be in the reversed order for uplink transmission of signals to the central station.

Following the functions 26 and 28, the resulting rate matched and interleaved signals are segmented for radio frames and physical channels in segmentation blocks 30 and 32 respectively to produce the signals for multiplexing by the multiplexer 10. Signals output by the multiplexer 10 are interleaved by a second interleaver 34 the outputs of which are segmented and mapped to dedicated physical channels in a segmentation and mapping block 36 for communications via a CDMA radio communications path in known manner.

As described in the patent application by Wen Tong et al. referred to above, the first interleaver 28 can have a performance that is sufficiently good to enable the second interleaver 34 to be omitted or reduced to a simple shuffling operation, for example as described below. This is desirable in particular because otherwise the second interleaver 34 has the potential to degrade the interleaving performed by each first interleaver 28, whereas each first interleaver 28 can be optimized for its particular rate matched data stream and QoS.

Accordingly, the first interleaver 28 is implemented as an algebraic interleaver providing a good random spreading property. The multiple encoded bit blocks or data transport frames for each QoS channel are mapped into a 2-dimensional matrix and are subjected to linear congruential rules to permute the rows and columns of the matrix to

implement the interleaving function. A maximum interleaving depth and time span can be determined by searching a set of best parameters. The interleaver consequently has a relatively simple form without disadvantages of known interleavers, such as requiring large memory sizes for look-up tables or inadequately accommodating the rate matching function.

Although the following description refers to rows and columns of a matrix, it should be understood that this is for convenience and clarity, that the rows and columns can be interchanged without changing the function of the interleaver, and that in practice and as described below the interleaver can operate by equivalent control of read or write addressing of memory locations of a linear memory in which data bits are stored, without any actual movement of the stored bits among the memory locations.

The interleaver 26 as described in the patent application referred to above operates to implement the following three steps:

1. Represent a number  $N_c$  of encoded blocks of data bits each of length  $N_r$  data bits as a matrix of  $N_r$  rows and  $N_c$  columns.
2. Permute the rows and columns of the matrix in accordance with:

$$\text{Row Permutation} \quad I_r(k) = [\alpha_r k + f_c(l)] \bmod N_r$$

$$\text{Column Permutation} \quad I_c(l) = [\alpha_c l + f_r(k)] \bmod N_c$$

where  $I_r(k)$  represents a data bit with a row index  $k$ ,  $k$  is an integer from 1 to  $N_r$ ,  $\alpha_r$  is a row permutation parameter and is an integer,  $f_c(l)$  is a positive function of a column index  $l$ ,  $l$  is an integer from 1 to  $N_c$ ,  $I_c(l)$  represents a data bit with the column index  $l$ ,  $\alpha_c$  is a column permutation parameter and is an integer,  $f_r(k)$  is a positive function of the row index  $k$ , and  $\bmod N_r$  and  $\bmod N_c$  represent modulo- $N_r$  and modulo- $N_c$  arithmetic respectively.

3. Derive interleaved data bits from the matrix column by column.

Step 1 can be slightly modified to accommodate different numbers of data transport frames with a given number of columns of the matrix. For example the matrix can have  $N_c = 8$  columns for  $N_c / \gamma$  data transport frames where  $\gamma = 1, 2, 4$ , or 8, the matrix having  $N_r / \gamma$  rows accordingly, with a consequent modification of step 3 to read out  $\gamma$  columns of the matrix per radio frame accordingly. For simplicity in the description below, it is assumed that  $\gamma = 1$  with  $N_c = 8$ .

For step 2, the row permutation parameter  $\alpha_r$  is chosen to be the largest prime number less than  $\lfloor N_r / \log_2(\log_2(N_r)) \rfloor$ , the column permutation parameter  $\alpha_c$  is chosen to be the largest prime number less than  $\lfloor N_c \rfloor$ , the function  $f_c(l) = ml + [N_r + 1] \bmod 2$ , where  $m$  is an integer equal to  $\lceil N_r / N_c \rceil$ , and the function  $f_r(k) = 2k + [N_c + 1] \bmod 2$ . The symbols  $\lfloor \cdot \rfloor$  refer to rounding down to an integer, and the symbols  $\lceil \cdot \rceil$  refer to rounding up to an integer. It can be appreciated that  $[N_r + 1] \bmod 2$  is zero when  $N_r$  is odd and is one when  $N_r$  is even, and that  $[N_c + 1] \bmod 2$  is zero when  $N_c$  is odd and is one when  $N_c$

is even, so that these parts of the functions  $f_c(l)$  and  $f_r(k)$  are simply the addition of one when the respective number  $N_r$  or  $N_c$  is even.

As indicated above, the rate matching punctures (deletes) redundant data bits (which are present as a result of the FEC encoding blocks 22 in the event that the data transport frame size is larger than the radio frame size, a maximum puncturing ratio being 20% of the transport frame size. Conversely, if the data transport frame size is smaller than the radio frame size, bits of the transport frame are repeated to achieve the rate matching. The rate matching is desired as far as possible to maximize the separation distances between the punctured bits and to equalize the number of punctured bits in each radio frame, i.e. to distribute the punctured bits uniformly among the radio frames with maximum separation.

In the case where the rate matching block 26 precedes the channel interleaver 28 as shown in Fig. 1, a known rate matching method as shown in Fig. 2 can be used.

Referring to Fig. 2, for each radio frame of segmentation size  $N_i$  bits, in a block 40 an integer  $y$  is determined as  $y = N_r - N_i$ ,  $y$  being greater than zero (positive) in the event that puncturing is required, being less than zero (negative) in the event that repetition of  $|y|$  bits is required, and being zero if no puncturing or repetition is required, a stop block 41 being reached in the latter event. As the steps shown in Fig. 2 are substantially the same for bit repetition ( $y < 0$ , shown at the right in Fig. 2) as for puncturing ( $y > 0$ , shown at the left in Fig. 2, except for the use of  $|y|$  and repetition instead of  $y$  and puncturing, only the case of puncturing is described in detail below.

If  $y > 0$ , puncturing of  $y$  of the  $N_r$  bits of the transport frame is required to produce the  $N_i$  bits of the radio frame. In this case in a block 42 a parameter  $e$  is initialized to a starting offset  $e_{os}$  which is determined in any desired manner for the particular radio frame, and a row counter  $r$  is initialized to 1. In a block 43, it is determined whether  $r \leq N_r$  and, if so, in a block 44 the value of  $e$  is reduced by  $2y$ . In a subsequent decision block 45 it is determined whether  $e \leq 0$ , and if so the bit in the row  $r$  is punctured at a block 46, the value of  $e$  is increased by  $2N_r$  in a block 47, the row counter  $r$  is increased by 1 in a block 48, and a return is made to the decision block 43. A negative decision at the block 45 (i.e.  $e > 0$ ) results in a return to the block 43 via the block 48 to increment the row counter  $r$  without any puncturing or change in the value of  $e$ . A negative decision at the block 43 (i.e.  $r > N_r$ ) indicates that the end of the frame has been reached, and accordingly the sequence ends at the stop block 41.

However, in the case where the rate matching block 26 follows the channel interleaver 28, the rate matching is carried out on the permuted (interleaved) bit stream and the problem of rate matching is considerably more complicated. Generally, the requirements of the channel interleaving and rate matching processes are not consistent.

More particularly, the design of an appropriate, and desirably optimized, rate matching pattern of punctured or repeated bits within the matrix of bits after the channel interleaving process represents a very complex or impractical task. This invention avoids this problem by providing an appropriate, and desirably optimized, rate matching pattern of punctured or repeated bits for the matrix before interleaving, and to use a de-interleaving or decoding process to determine corresponding bits to be punctured or repeated at the output of the channel interleaver. This process is facilitated by the fact that the de-interleaving, or decoding, process can be implemented by exactly the same structure as the interleaving process, as further described below. For convenience and clarity, the following description refers to the matrix of bits before interleaving (or after de-interleaving) as the natural matrix NM, and to the matrix of bits after interleaving as the randomized matrix RM.

Fig. 3 illustrates an implementation of a channel interleaver 28 and rate matching block 26 in accordance with an embodiment of this invention. As illustrated in Fig. 3, the interleaver 26 includes a working memory 50 with two halves, alternately used in known manner for writing into and reading from the memory, each for storing the  $N_r N_c$  data bits represented in the matrix as described above, these data bits being written into the memory linearly corresponding to the row-by-row organization of the matrix. A modulo- $N_r$  row counter 51 is responsive to a clock signal CLK to provide a count representing the row index  $k$ , and a carry output of this counter 51 is supplied to a modulo- $N_c$  column counter 52 to provide a count representing the column index  $l$ . The counts  $k$  and  $l$  of the counters 51 and 52 are supplied to an address encoder 53 shown within a dashed line box in Fig. 3. More specifically, the count of the column counter 53 is supplied to multipliers 54 and 55 which are also supplied with the parameters  $\alpha_c$  and  $m$  respectively to produce products representing  $\alpha_c l$  and  $ml$  respectively, and the count of the row counter 51 is supplied to multipliers 56 and 57 which are also supplied with the integer 2 and the parameter  $\alpha_r$  respectively to produce products representing  $2k$  and  $\alpha_r k$  respectively. An adder 58 adds the outputs of the multipliers 54 and 56 and selectively adds 1 or 0 depending upon whether  $N_c$  is even or odd respectively, and the output of the adder 58 is reduced to modulo- $N_c$  form by a modulo function 59 to complete the column permutation function described above. An adder 60 adds the outputs of the multipliers 55 and 57 and selectively adds 1 or 0 depending upon whether  $N_r$  is even or odd respectively, and the output of the adder 60 is reduced to modulo- $N_r$  form by a modulo function 61 to complete the row permutation function described above. Each of the modulo functions 59 and 61 can comprise comparison and subtraction functions. Outputs of the functions 59 and 61 are combined in a read address combiner 62 to produce an address for reading the respective data bit in its interleaved sequence from the memory 50. As illustrated in Fig.

3, the read address is supplied to the memory 50 via a switch 63 which is provided as described below.

If the number of rows  $N_r$  is a power of two, then the address combiner 62 can simply combine the output of the modulo function 61 as the least significant bits, and the output of the modulo function 59 as the most significant bits, of the read address for the memory 50; equivalently the output of the function 61 is added by the address combiner 62 to  $N_r$  times the output of the function 59.

It may be desired to interleave data bits in arbitrary-sized frames that are not an integer multiple of  $N_c$ . In this case, the number of rows of the matrix is selected to accommodate all of the data bits to be interleaved, and the last few (less than  $N_c$ ) memory locations in the working memory 50 are not written into. In order to omit the data bits of these memory locations from the interleaved data bits, the interleaver 28 of Fig. 3 also includes a decoder 64 which detects these memory locations in the read address output of the address combiner 62, and upon such detection opens the switch 63 to prevent reading of data from the memory 50 in respect of these locations. In order to provide a constant data output rate of the interleaved data bits from the memory 50, the interleaver 28 of Fig. 3 further includes a FIFO (first-in, first-out) memory 65, clocked by the clock signal CLK, via which the interleaved data bits are supplied to an output line 66 of the interleaver, the FIFO 65 being pre-filled at the start of each interleaving operation and having a size (e.g. up to  $N_c$ ) sufficient to allow for the non-read, and hence omitted, memory locations.

The interleaved data bits on the line 66 are supplied to the rate matching block or function 26 also illustrated in Fig. 3. This rate matching function comprises a rate matching address generator 70 which is also supplied with the clock signal CLK, an address separator 71, an address decoder 72, a buffer or store 73, comparators 74, and a data bit selector 75 providing a rate matched data output on a line 76. Similarly to the provision of the FIFO 65 for providing a constant data bit rate output of the interleaver 28, the rate matching function 26 may also include a FIFO or other buffer (not shown) for providing a constant rate of data bits from the output line 76.

The rate matching address generator 70 generates at its output, as described further below, the address in the natural matrix NM of each punctured or repeated bit in accordance with the puncturing or repetition pattern which is determined for this process. This address is separated into most significant and least significant components by the address separator 71, whose operation is inverse to that of the read address combiner 62 described above. Thus if the number of rows  $N_r$  is a power of two, then the address separator 71 can simply separate the address bits output by the generator 70 into most significant bits and least significant bits; equivalently the address from the generator 70 is

divided by  $N_r$  to produce an integer quotient and a remainder which constitute the two outputs of the address separator 71.

The address decoder 72 performs the inverse function of the address encoder 53. As indicated above, with the algebraic interleaving process described herein the structure of the de-interleaver can be exactly the same as the structure of the interleaver, and correspondingly the address decoder 72 is exactly the same as the address encoder 53. Accordingly, the detailed structure of the address decoder 72 is not illustrated in Fig. 3, it being identical to the structure of the address encoder 53 as shown in Fig. 3. It can be appreciated that this same-structure characteristic of the complementary operations for interleaving and de-interleaving provides a substantial advantage and simplification in implementing these functions.

The outputs of the address decoder 72 are buffered in the store 73, and the buffered outputs from the store are compared in the comparators 74 with the current counts  $k$  and  $l$  of the row counter 51 and column counter 52 respectively of the channel interleaver 28, to provide a selector control signal on a line 77 with a predetermined state when the compared values are the same. The selector control signal is thus produced on the line 77 with this state at each time that a bit on the line 66 is to be punctured or repeated. At other times, for bits which are not punctured or repeated, the control signal on the line 77 controls the selector 75 to supply to its output line 76, synchronously as determined by the clock signal CLK, a bit from the line 66 supplied to a middle one (as shown in Fig. 3) of three inputs of the selector 75. At the time of each bit to be repeated or punctured, the control signal on the line 77 instead controls the selector 75 to supply to its output line a bit from either its upper input or its lower input (as shown in Fig. 3), depending upon whether bits are to be repeated or punctured respectively, as determined by a further control input P/R to the selector 75. The upper input of the selector 75 is connected to the output line 76 to provide for bit repetition, and the lower input of the selector 75 is illustrated as having no connection to provide for bit puncturing. As indicated above, a constant output data bit rate for the interleaved and rate matched data bits is provided by a buffer (not shown) to which data bits on the output line 76 are supplied.

Because of the address decoding provided by the decoder 72 in the rate matching function 26, the rate matching address generator 70 can simply determine the desired pattern of punctured or repeated bits in terms of the normal matrix addresses in the manner described above with reference to Fig. 2, using the single parameter  $e_{os}$  determined in a desired manner to optimize this pattern. For example, this parameter could be determined by an equation such as  $e_{os} = [2py + 1] \bmod 2N_r$ , where as described above  $y$  is the number of bits to be punctured or repeated for each column of the matrix, and  $p$  is a column index from 0 to 7 (for the case of  $N_c = 8$ ).



This example, with interleaving as described above of 8 data transport frames each of 10 bits, and requiring a maximum puncturing ratio of 20% to produce channel interleaved and rate matched radio frames each of 8 bits (a total of 16 out of 80 bits being punctured or deleted), is further illustrated by the following Tables 1, 2, and 3. Thus  $N_c = 8$  and  $N_r = 10$ . Table 1 illustrates entry of the 80 data bits, numbered 0 to 79, row by row in a 10 by 8 natural matrix with the row index  $k$  from 1 to 10 and the column index  $l$  from 1 to 8:

		1							
		1	2	3	4	5	6	7	8
k	1	0	1	2	3	4	5	6	7
	2	8	9	10	11	12	13	14	15
	3	16	17	18	19	20	21	22	23
	4	24	25	26	27	28	29	30	31
	5	32	33	34	35	36	37	38	39
	6	40	41	42	43	44	45	46	47
	7	48	49	50	51	52	53	54	55
	8	56	57	58	59	60	61	62	63
	9	64	65	66	67	68	69	70	71
	10	72	73	74	75	76	77	78	79

Table 1

The channel interleaving as described above produces a randomized matrix as shown by the following Table 2:

		1							
		1	2	3	4	5	6	7	8
k	1	57	40	79	62	45	28	11	74
	2	35	18	1	64	23	6	69	52
	3	13	76	59	42	25	8	47	30
	4	71	54	37	20	3	66	49	32
	5	73	56	15	78	61	44	27	10
	6	51	34	17	0	39	22	5	68
	7	29	12	75	58	41	24	63	46
	8	7	70	53	36	19	2	65	48
	9	9	72	31	14	77	60	43	26
	10	67	50	33	16	55	38	21	4

Table 2

The rate matching as described above then punctures 16 bits, 2 from each column of the randomized matrix, in a pattern produced by the rate matching algorithm to give a punctured randomized matrix as shown by the following Table 3:

		1							
		1	2	3	4	5	6	7	8
k	1	57	40	79	62	45	28		74
	2	35	18	1		23	6	69	52
	3	13	76	59	42		8		30
	4	71		37	20	3	66	49	
	5	73	56	15	78		44	27	10
	6	51		17	0	39	22	5	
	7		12		58	41	24	63	46
	8	7	70	53	36	19		65	48
	9		72		14	77	60	43	26
	10	67	50	33		55		21	4

Table 3

The channel interleaved and rate matched data bits are derived column by column from Table 3, i.e., with the order [ 57, 35, ..., 51, 7, 67, 40, ..., 26, 4 ]. The punctured bits are 2, 9, 11, 16, 25, 29, 31, 32, 34, 38, 47, 54, 61, 64, 68, and 75, for which the maximum puncture distance is 9 (25 - 16) and the minimum puncture distance is 1 (32 - 31); this small minimum puncture distance indicates that this particular example is not optimum, a larger minimum puncture distance being desirable. It can be appreciated that numerous other determinations of the parameters, and in particular of the parameter  $e_{os}$ , can be provided to optimize the puncturing process.

As indicated above, it is desirable for operation of the second interleaver 34 not to degrade the performance achieved as a result of the first interleaver 28, and to this end it is advantageous for the second interleaver 34 to be reduced to a simple shuffling operation, which interleaves data streams with different QoS while retaining the spreading properties achieved by the first interleaver 28 for each QoS data stream.

Fig. 4 shows a flow chart of a bit shuffling algorithm which can be used advantageously to interleave bits of two data streams of interleaved radio frames provided as described above from respective service blocks 12 provided via the service multiplexer 10 in Fig. 1. Denoting one stream  $TQ_1$  having frames of  $N_1$  bits and a second stream  $TQ_2$  having frames of  $N_2$  bits, with  $N_1 \geq N_2$ , Fig. 4 illustrates how bits of the stream  $TQ_2$  are inserted into the stream  $TQ_1$ .

Referring to Fig. 4, initially in a block 82 a parameter  $e$  is initialized to  $N_1$  and a counter  $r$  is initialized to 1. In a block 83, it is determined whether  $r \leq N_1$  and, if so, in a

block 84 the value of  $e$  is reduced by  $2N_2$ . In a subsequent decision block 85 it is determined whether  $e \leq 0$ , and if so at a block 86 the next bit in the stream  $TQ_2$  is inserted into the stream  $TQ_1$ , the value of  $e$  is increased by  $2N_1$  in a block 87, the counter  $r$  is increased by 1 in a block 88, and a return is made to the decision block 83. A negative decision at the block 85 (i.e.  $e > 0$ ) results in a return to the block 83 via the block 88 to increment the counter  $r$  without any bit insertion or change in the value of  $e$ . A negative decision at the block 83 (i.e.  $r > N_1$ ) indicates that the end of the frame has been reached, and accordingly the sequence ends at a stop block 81.

For more than two data streams, the same process is applied recursively for the successive data streams. It can be appreciated from the above description and the illustration in Fig. 4 that the steps of this process have a direct correlation with the steps of the puncturing and repetition processes of Fig. 2, so that implementation of this recursive shuffling process can be particularly convenient.

As indicated above, the puncturing of bits to achieve the desired rate matching is applied to data bits which have redundancy due to the FEC encoding provided by the encoders 22. One preferred form of encoding is so-called turbo (parallel concatenated convolutional) coding, in which the encoded data bits comprise the input data bits themselves, referred to as systematic data bits  $S$ , and parity bits  $P1$  and  $P2$  provided by convolutional coders operating on the input data bits and on interleaved input data bits. The parity bits  $P1$  and  $P2$  are typically punctured within the turbo coder to provide a desired rate turbo coder. For encoders 22 constituted by turbo coders, it is necessary to ensure that the subsequent rate matching function 26 does not puncture any of the systematic bits  $S$ , but only the parity bits  $P1$  and/or  $P2$ . In the case of repetition, it has been determined that repetition of the parity bits  $P1$  and  $P2$  by a factor of the order of 2 or 3 times the repetition of the systematic bits  $S$  provides a performance gain.

To these ends, Fig. 5 illustrates a modification of part of the arrangement of Fig. 1 for channel interleaving and rate matching of data encoded by turbo coding. Referring to Fig. 5, a turbo coder constituting one of the FEC encoders 22 is shown within a dashed line box 90 and, as is well known, comprises a turbo code interleaver 91 which interleaves input data bits, and two convolutional coders 92 which operate on the input data bits before and after interleaving to produce parity bits  $P1$  and  $P2$ , the input data bits also being supplied to the coder outputs as systematic bits  $S$ . A puncturing block, not shown, may also be present to select only some of the parity bits  $P1$  and  $P2$  for supply to the coder outputs.

Instead of a single channel interleaver as described above, Fig. 5 illustrates that individual channel interleavers 93 are provided for the systematic bit stream and the parity bit stream. As shown in Fig. 5, there are three channel interleavers 93, but it can be appreciated that the streams of parity bits  $P1$  and  $P2$  can be combined and interleaved

together, so that only two channel interleavers are provided, one for the systematic bit stream and the other for the parity bit stream. Further inputs to the channel interleavers 93 in Fig. 5 indicate the multiplexing of systematic and parity bit streams, respectively, for multiple channels, corresponding to the multiplexer 24 in Fig. 1.

The rate matching function, which follows the channel interleavers 94, is shown within a dashed line box 94. A puncturing function 95 is applied only to the channel-interleaved parity bit streams, whereas a repetition function 96 can be provided to the parity and systematic bit streams, a selector 97 being illustrated to couple the channel-interleaved bits accordingly. The puncturing and repetition can be as described above. It can be appreciated that the illustration in Fig. 5 in this respect is intended diagrammatically to represent the principle that puncturing is not applied to the systematic bits, rather than showing an actual implementation of the rate matching function. It can be appreciated, for example, that puncturing or repetition, as required, could be applied only to the parity bit streams to provide the desired rate matching, without any puncturing or repetition of the systematic bit stream.

Although the above description refers to separate functions and units for the various processes described herein, it can be appreciated that these can in many cases be implemented using functions of one or more digital signal processors or other integrated circuits.

Although particular embodiments and examples of the invention have been described above, it can be appreciated that numerous modifications, variations, and adaptations may be made without departing from the scope of the invention as defined in the claims.

#### 4. Brief Description of Drawings

The invention will be further understood from the following description with reference to the accompanying drawings, in which:

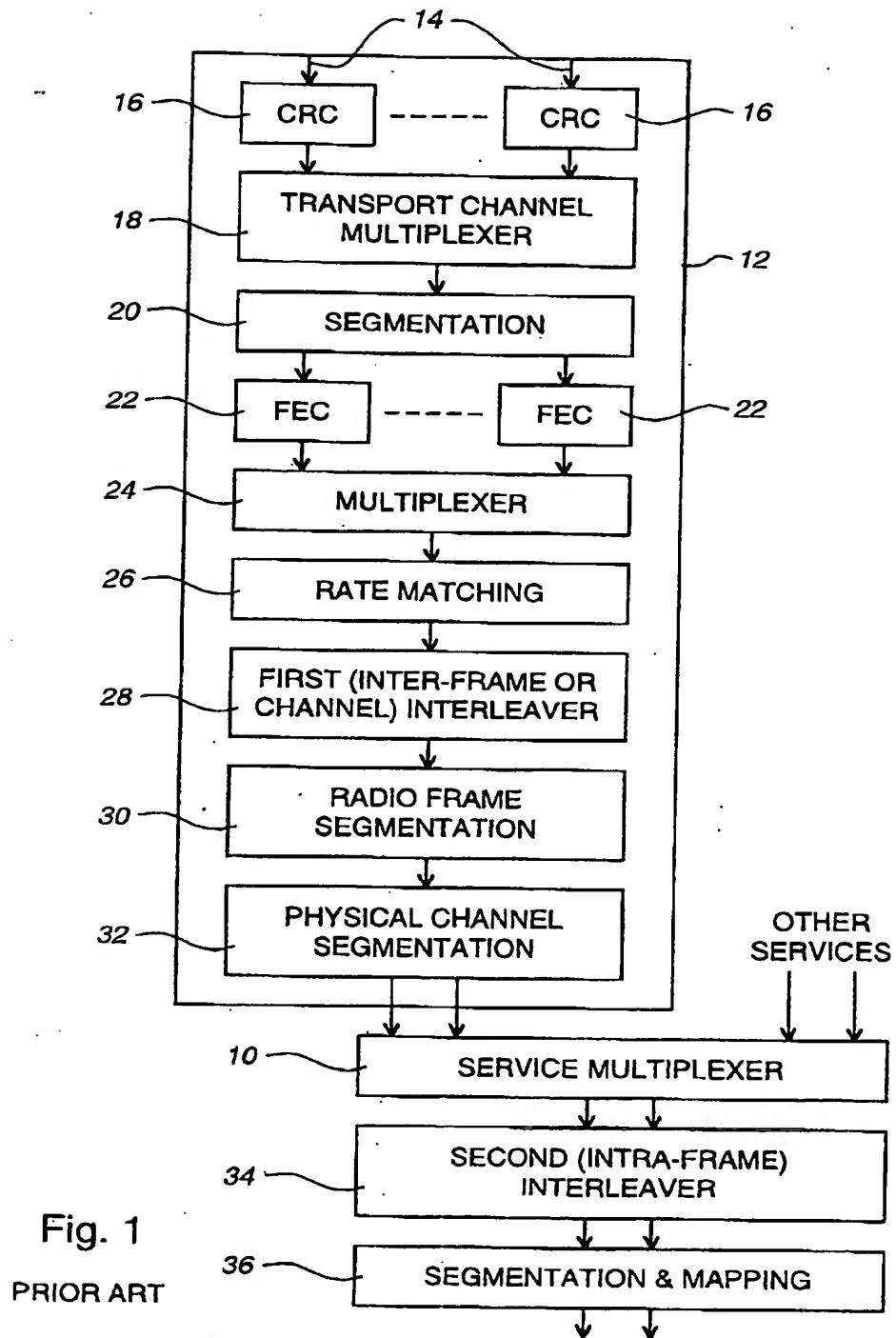
Fig. 1 illustrates a known arrangement for service multiplexing and channel interleaving in a 3rd generation CDMA communications system;

Fig. 2 is a flow chart relating to a known rate matching algorithm;

Fig. 3 illustrates an implementation of an interleaver and a rate matching arrangement in accordance with an embodiment of this invention;

Fig. 4, which is on the same sheet as Fig. 2, is a flow chart relating to shuffling for a second stage of interleaving in the arrangement of Fig. 1; and

Fig. 5 illustrates a modification of part of the arrangement of Fig. 1 for channel interleaving and rate matching of data encoded by turbo (parallel concatenated convolutional) coding.



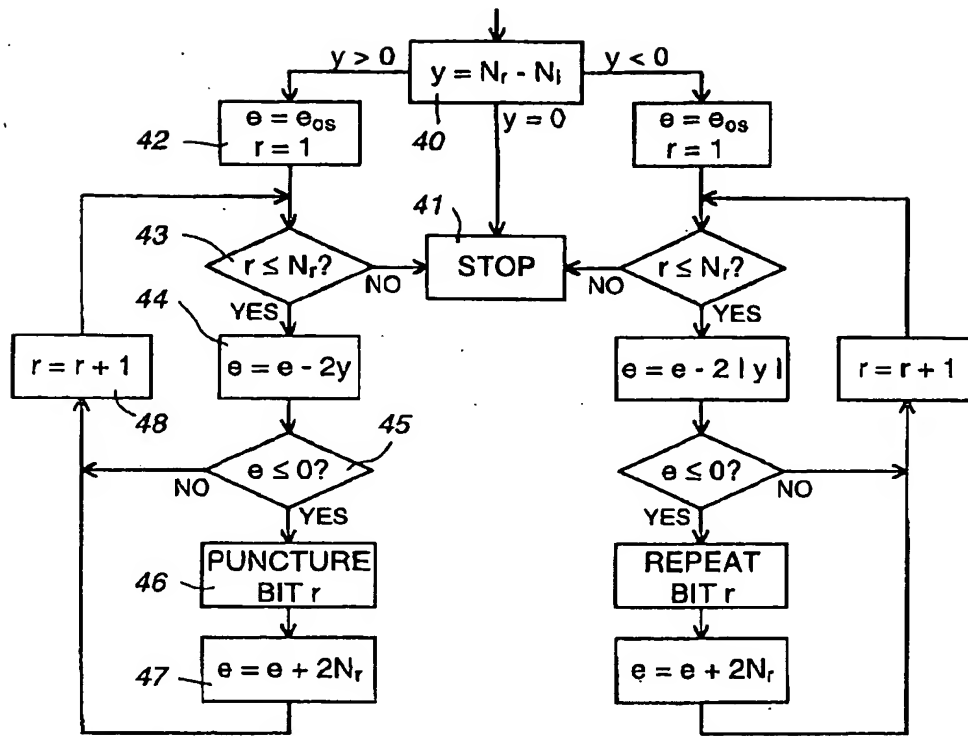
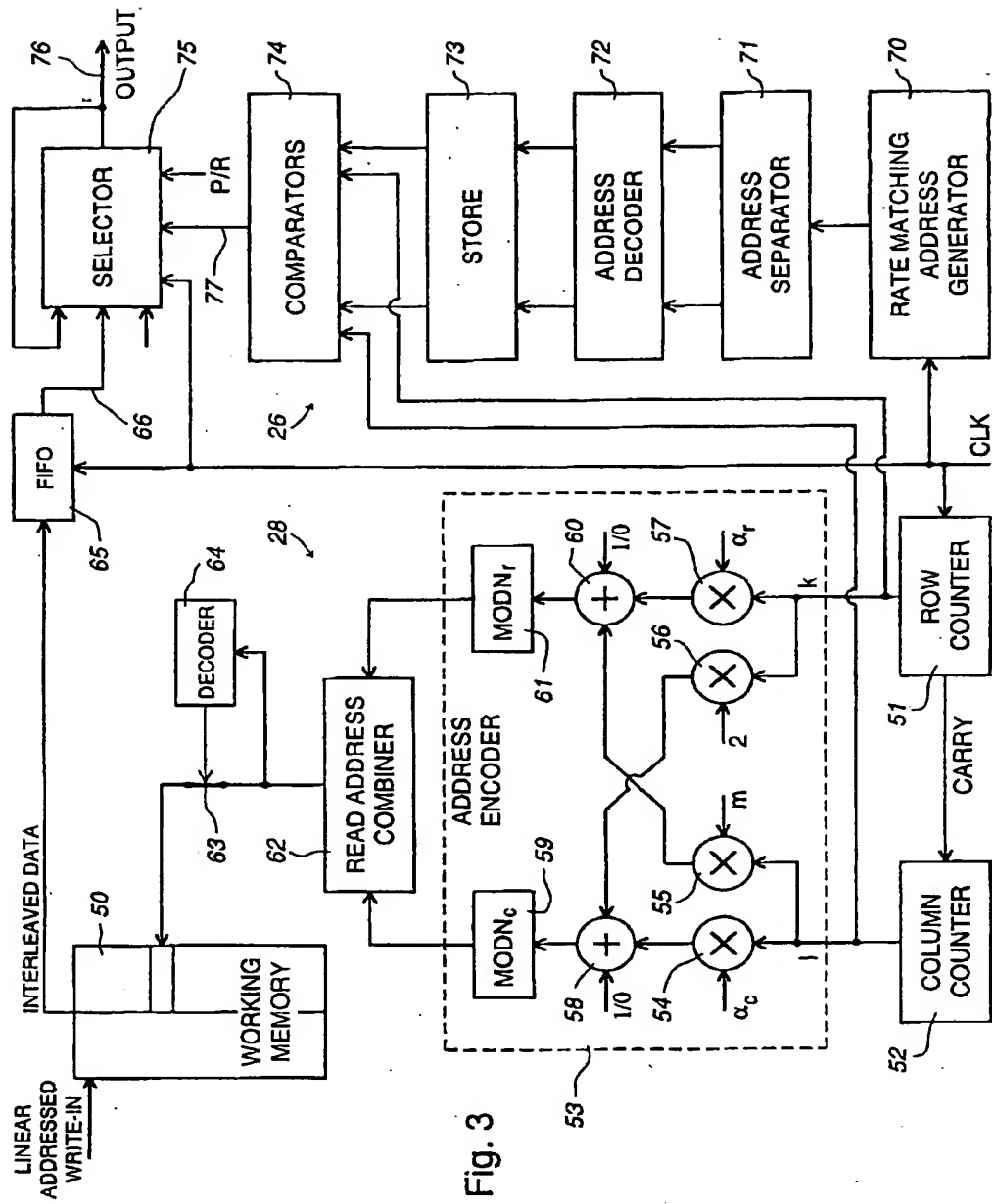


Fig. 2

PRIOR ART



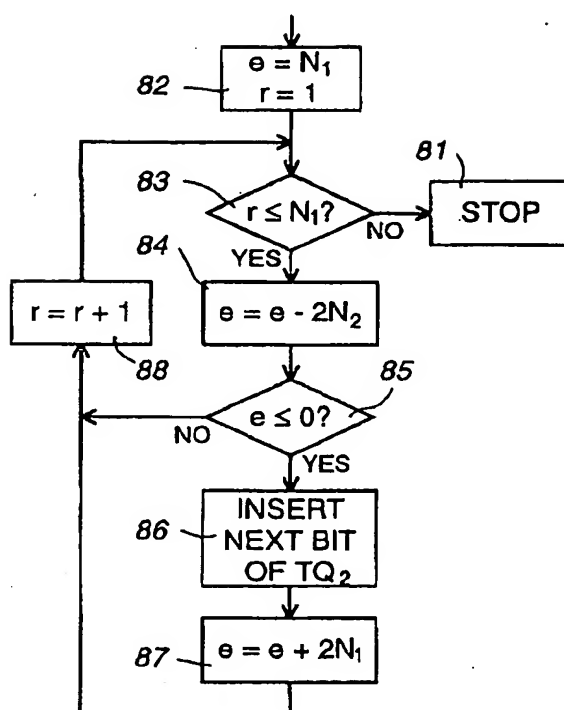
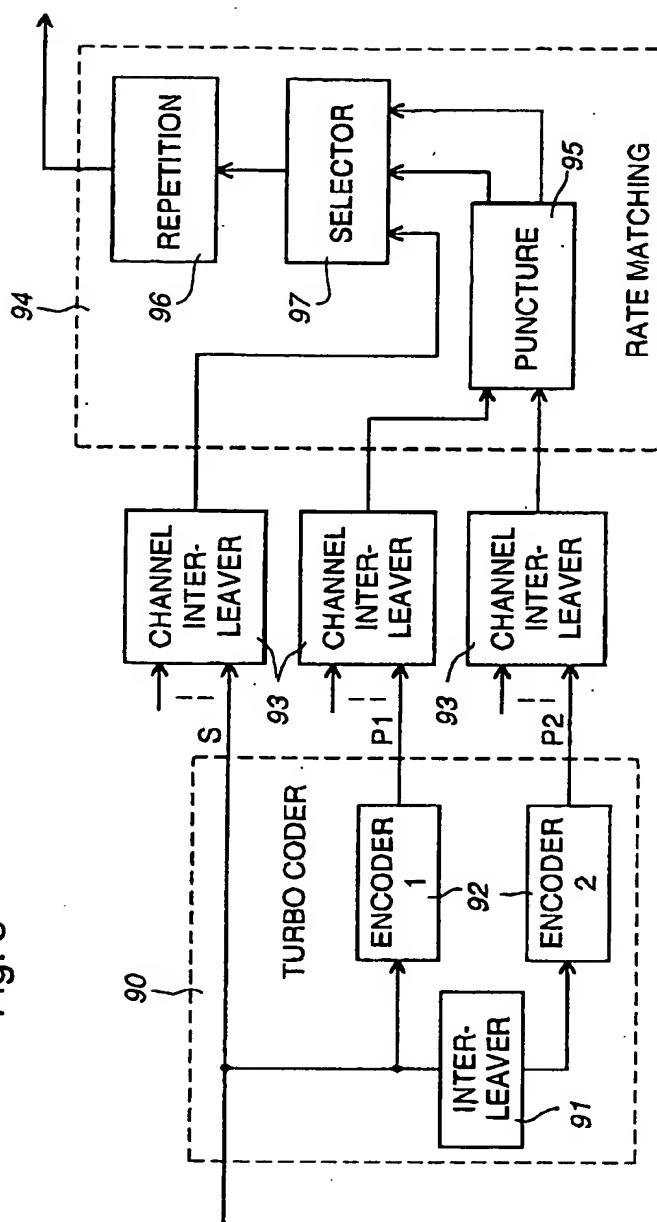


Fig. 4



Fig. 5



A method of and apparatus for matching a rate of data bits, in a matrix of data bits interleaved by a predetermined interleaving process, to a desired rate by deletion of redundant data bits or repetition of data bits derived from the matrix, includes steps of determining in a non-interleaved matrix of the data bits a pattern of bits to be deleted or repeated to provide the desired data rate, decoding an address of each bit in said pattern in a manner inverse to the interleaving process to produce a respective address of the bit in the matrix of interleaved data bits, and deleting or repeating the respective bit in the interleaved data bits in dependence upon the respective address. The address decoding is performed in the same manner as a coding of addresses for producing the interleaved data bits from the non-interleaved matrix of the data bits. The specification also discloses an advantageous interleaving process for channel interleaving in a 3rd generation CDMA wireless communications system, a shuffling method for a second stage of interleaving in such a system, and how the rate matching can be conveniently applied to turbo-coded data coded.

## 2. Representative drawing

Fig. 3